J Glob Optim (2006) 36:365–377
DOI 10.1007/s10898-006-9014-3

ORIGINAL ARTICLE

Bayesian stopping rules for greedy randomized procedures

Carlotta Orsenigo · Carlo Vercellis

Received: 13 August 2004 / Accepted: 21 February 2006 / Published online: 4 July 2006

Abstract A greedy randomized adaptive search procedure (GRASP) is proposed for the approximate solution of general mixed binary programming problems (MBP). Examples are provided of practical applications that can be formulated as MBP requiring the solution of a large number of problem instances. This justifies, from both a practical and a theoretical perspective, the development of stopping rules aimed at controlling the number of iterations in a GRASP. To this end, a bayesian framework is laid down, two different prior distributions are proposed and stopping conditions are explicitly derived in analytical form. Numerical evidence shows that the stopping rules lead to an optimal trade-off between accuracy and computational effort, saving from unneeded iterations and still achieving good approximations.

Keywords GRASP · Bayesian stopping rules · Heuristics · Mixed binary programming

1 Introduction

Greedy randomized adaptive search procedures (GRASP) are metaheuristics that have been successfully applied to a broad collection of difficult optimization problems, reviewed in (Feo and Resende 1995; Festa and Resende 2002; Resende and Ribeiro 2003); see also (Betrò and Vercellis 1986) for one of the earliest formulations of GRASP. In its basic version, a GRASP is an iterative multi-start Monte Carlo algorithm in which randomization steps are introduced as improvements over the myopic selection criteria of a typical greedy heuristic in order to generate alternative feasible solutions. The best observed value in a sequence of independent

C. Orsenigo
Dipartimento di Scienze Economiche, Aziendali e Statistiche, Università di Milano, Via Conservatorio 7 Milano 20122, Italy
e-mail:carlotta.orsenigo@unimi.it

C. Vercellis (✉)
Dipartimento di Ingegneria Gestionale, Politecnico di Milano, P.za Leonardo da Vinci 32, Milano 20133 Italy
e-mail:carlo.vercellis@polimi.it

runs is then retained as an approximation to the optimum value. Additional refinements have been proposed to improve the outlined framework. For instance, at each iteration the random solution achieved at the end of the generation phase can be improved through a local search phase, in which a suitable neighborhood is systematically explored seeking for a better solution. Other strategies were aimed at incorporating some form of learning mechanism in the memoryless structure of GRASP (Prais and Ribeiro 2000), or introduced cost perturbations for problems where no obvious greedy algorithms are available (Canuto et al. 2001).

It appears that investigations on how large the number of different random solutions should be were not addressed within the rich body of literature on GRASP, and to a large extent the problem of determining the sample size for the Monte Carlo repeated trials was left to empirical subjective assessment. Although somewhat unsatisfactory as a theoretical perspective, this pragmatic approach may seem appropriate for those optimization problems for which each GRASP iteration is very fast, and a single instance has to be solved as a one-shot problem. There are however situations that involve the solution of a very large number of instances, up to thousands or more, therefore requiring a careful and regulated stopping rule to balance accuracy and computational effort. For instance, in the construction of oblique classification trees in learning theory (Orsenigo and Vercellis 2003, 2004a,b) a large collection of mixed binary programming problems has to be solved, as sketched in Sect. 2. In industrial applications, also considered in more details in Sect. 2, production planning problems with minimum lot size constraints can be formulated as mixed binary programming, and have to be solved repeatedly for different product lines. Finally, similar requirements arise when optimization is used in connection to simulation models, and each objective function evaluation during the optimization process requires a long simulation run to be performed, so that the user is interested in keeping low the number of iterations. Notice that the problem of deriving stopping rules for multi-start algorithms was addressed for global optimization by a number of authors (Betrò and Vercellis 1986; Boender and Rinnooy Kan 1987; Boender et al. 1987; Betrò and Schoen 1987, 1992; Hart 1999).

In this paper we first describe families of optimization problems that arise in practical applications and require the solution of a large number of instances. It will be seen that they can be formulated as mixed binary programming problems. Then, a class of GRASP for their approximate solution is proposed. In Sect. 3 we develop a bayesian framework for devising sequential stopping rules for GRASP, discussing alternative assumptions concerning the prior distribution, and showing how the posterior can be analytically calculated given the sample. Finally, computational tests are presented in Sect. 4 to assess the potential advantages of the proposed approach.

2 GRASP for mixed binary programming problems

In this section we propose a GRASP for solving the general mixed binary programming problem (MBP), formulated as

$$\min \quad z(\mathbf{x}, \mathbf{y}) = \mathbf{c}'\mathbf{x} + \mathbf{h}'\mathbf{y}, \tag{MBP}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{x} + \mathbf{D}\mathbf{y} = \mathbf{b}, \tag{1}$$

$$\mathbf{x} \geq \mathbf{0}, \mathbf{0} \leq \mathbf{y} \leq \mathbf{1}, \mathbf{y} \text{ integer},$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^q$ are vectors of continuous and binary variables; $\mathbf{c} \in \mathbb{R}^n, \mathbf{h} \in \mathbb{R}^q, \mathbf{b} \in \mathbb{R}^m, \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{D} \in \mathbb{R}^{m \times q}$ are vectors and matrices of parameters. Let $(\mathbf{x}^*, \mathbf{y}^*)$ and z^* denote respectively the optimal solution and the optimum value associated to MBP,

whenever they exist. A remarkable number of applied optimization problems can be naturally expressed in the form MBP. In particular, we are interested here in situations that require a large number of instances of MBP to be solved, because in similar cases the stopping conditions developed in Sect. 3 would play a critical role. Therefore, a brief description of two applied problems requiring the solution of large collections of MBP instances will be provided, before developing a GRASP for MBP.

2.1 Example 1: MBP arising in learning theory

In the field of learning theory, some classification models based on discrete variants of support vector machines were recently proposed and formulated as MBP (Orsenigo and Vercellis 2003, 2004a,b). In a binary classification problem we are provided with a set of m points defined in the n-dimensional space $\mathbb{R}^n$ and represented by a $m \times n$ matrix $\mathbf{A}$. The membership of each point to one of the two classes, labeled as $\{+1\}$ and $\{-1\}$, can be specified by a given $m \times n$ diagonal matrix $\mathbf{D}$ with ones or minus ones along its main diagonal. To discriminate points belonging to class $\{+1\}$ from those labeled as $\{-1\}$ a linear hyperplane can be derived by minimizing a suitable measure of inaccuracy. Let $\mathbf{w} \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}$ denote the coeffcients of the separating hyperplane to be determined. Elements of statistical learning theory, described in the papers quoted above and in the references therein, lead back the evaluation of $(\mathbf{w}, \gamma)$ to the minimization of a weighted sum of three terms: the reciprocal of the generalization capability of the discriminating model, its misclassification error and its complexity. Let $\mathbf{y}$ be a vector of binary variables each taking the value one if and only if the corresponding point is misclassified, and $\mathbf{h}$ be the vector of misclassification costs. Let also $\mathbf{s}$ indicate a vector of binary variables each taking the value one if and only if the separating hyperplane has a non-zero coefficient along the corresponding dimension i, that is if $\mathbf{w}_i \neq 0$, and $\mathbf{k}$ be the vector of penalty costs aimed at inducing sparse vectors $\mathbf{w}$. Thus, the problem of deriving an optimal separating hyperplane of low complexity can be formulated as a MBP

$$\min \quad z(\mathbf{w}, \gamma, \mathbf{u}, \mathbf{y}, \mathbf{s}) = \beta_1 \mathbf{e}'\mathbf{u} + \beta_2 \mathbf{h}'\mathbf{y} + \beta_3 \mathbf{k}'\mathbf{s}, \tag{FDVM}$$

$$\text{s.t.} \quad \mathbf{D}(\mathbf{A}\mathbf{w} - \mathbf{e}\gamma) \geq \mathbf{e} - Q\mathbf{y}, \tag{2}$$

$$-\mathbf{u} \leq \mathbf{w} \leq \mathbf{u}, \tag{3}$$

$$\mathbf{u} \leq R\mathbf{s}, \tag{4}$$

$$\mathbf{u} \geq \mathbf{0}, \quad \mathbf{y} \in \{0,1\}^m, \quad \mathbf{s} \in \{0,1\}^n,$$

where $\mathbf{e}$ is a n-vector of ones; $\mathbf{u} \in \mathbb{R}^n$ is a vector of bounding variables; Q and R are sufficiently large constants; $\beta_1, \beta_2, \beta_3$ are parameters to control the trade-off among the objective function terms, representing respectively the reciprocal of the generalization capability, the accuracy and the complexity of the separating function. Model FDVM can be used as a linear perceptron or, alternatively, be framed within a recursive procedure for the generation of oblique classification trees, to derive an optimal separating hyperplane at each node of the tree. In this case, the need to solve a large number of instances of FDVM arises for multicategory classification tasks, usually formulated as a sequence of binary classification problems. In practice, to separate the points belonging to each node of the tree one has to determine at least L separating hyperplanes, where L is the number of different classes. Moreover, the parameters $\beta_1, \beta_2, \beta_3$ have to be exhaustively tuned in order to generate the best classification trees in terms of accuracy, generalization capability and complexity, and thus model FDVM has to be solved for a large number of combinations of $\beta_1, \beta_2, \beta_3$.

2.2 Example 2: MBP arising in production planning

In the context of production planning, binary variables are often needed to express conditions of minimum lot size on the continuous production quantities, or to model stepwise nonlinear cost functions, as in the case of fixed production costs. To provide a generic description of this class of problems, denote by **x** the vector of decision variables that represent the production volumes for each product type, plant and period. Let **y** be the vector of binary variables, each taking the value one if and only if the corresponding production quantity is greater than zero. Let also **c** be a vector of unit production costs and **h** a vector of fixed costs. The production planning problem can therefore be formulated as a MBP

$$\min \quad z(\mathbf{x}, \mathbf{y}) = \mathbf{c}'\mathbf{x} + \mathbf{h}'\mathbf{y}, \tag{PPL}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \tag{5}$$

$$\mathbf{x} \geq \mathbf{g}\mathbf{y}, \quad \mathbf{x} \leq M\mathbf{y}, \tag{6}$$

$$\mathbf{x} \geq \mathbf{0}, \quad \mathbf{0} \leq \mathbf{y} \leq \mathbf{1}, \mathbf{y} \text{ integer},$$

where **g** is a vector of minimum lot sizes, M is a big constant used to force the binary variables and constraints (5) incorporate different type of logical and physical conditions, such as plants and manpower capacity, balance on demand fulfillment, critical resources and components availability. In practice, one has to solve a large number of MBP instances, since problem PPL is usually decomposable in subproblems, by partitioning separate products, plants or periods. Additionally, even when no natural separation is evident, mathematical decomposition schemes can be applied to derive many smaller subproblems of the same MBP form, such as in (Fumero and Vercellis 1996, 1997).

2.3 A GRASP for MBP

Given an optimization problem, let A be an algorithm designed to generate a feasible solution whose objective function value z^A is close to the optimum z^*. Algorithm A is said *randomized* if some of the steps it performs depend on the outcomes of a random number generation process. Therefore, M repeated executions of A result in a sequence of independent realizations $z_i^A, i = 1, 2, \ldots, M$, of the random variable (r.v.) Z^A. The best value observed $\bar{z} = \min_{1 \leq i \leq M} z_i^A$ in a sequence of M independent runs is retained as an approximation to the optimum value. Hence $\bar{z}$ is a realization of the r.v. $\bar{Z}$. A GRASP metaheuristic is an iterative algorithm that performs a construction phase to build a good feasible solution according to a randomized variant of a greedy deterministic strategy. This means that for each step during the construction phase, allowable moves are ranked according to some myopic measure of attractiveness, and the move to be implemented is chosen randomly among a group of attractive ones, instead of picking the single highest ranked move as for the deterministic greedy algorithm. The construction phase is usually followed by an improvement phase, in which a local search is performed by iteratively searching a suitable neighborhood of the current best solution, until no locally better solutions can be found. The pseudo-code in Fig. 1 provides a general description of the GRASP we propose, while Fig. 2 sketches the local search procedure. The main novelty in our formulation lies in the stopping subprocedure controlling the while-loop, that in previous studies on GRASP was implemented by simply reaching a prefixed number of iterations, with the solely exception in (Betrò and Vercellis 1986).

In what follows we will detail the three main subprocedures contained in the GRASP pseudo-code, in order to solve MBP. For both examples described above, as for many practical

```
procedure GRASP;
  ReadData;
  k=0
  do
    k = k + 1;
    sol[k] = GreedyRandomizedBuild;
    sol[k] = LocalImprove(sol);
    UpdateSol(sol[k], bestsol);
  while (Stopping(sol, bestsol, k));
end GRASP.
```

Fig. 1 Pseudo-code for the GRASP metaheuristic

```
procedure LocalImprove(currsol);
  do
    find sol in Neighbor(currsol) with z(sol) < z(currsol);
    currsol = sol;
  while (currsol is not locally optimal);
end LocalImprove.
```

Fig. 2 Pseudo-code for the local search metaheuristic

applications, it is straightforward to generate a feasible solution. We therefore assume that a feasible solution $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ to MBP is known, or otherwise its existence has been ruled out. We denote by $Q^l = \{j : \hat{y}_j = l,\ j = 1, 2, \ldots, q\}, l \in \{0, 1\}$, the two sets of indices of binary variables assuming value 0 or 1 in the feasible solution. Let also $(\mathbf{x}^A, \mathbf{y}^A)$ be the feasible solution to MBP generated by the proposed GRASP, and $z^A = z(\mathbf{x}^A, \mathbf{y}^A)$ its objective function value.

The construction phase is performed as follows. First the continuous relaxation of MBP is solved, and let $z^{LP} = z(\mathbf{x}^{LP}, \mathbf{y}^{LP})$ be its optimal solution. Suppose that at least one binary variable assumes a fractional value in the relaxed solution, since otherwise we have reached also the optimal mixed binary solution and the procedure is stopped. In each single step of the construction phase we try to force to 0 one of the fractional binary variables. As a measure of attractiveness for the fractional variables, we consider the potential decrease in the objective function obtained by forcing a variable to zero, that is $\rho_j = h_j y_j^{LP}$. Let $\Psi = \{j : 0 < y_j^{LP} < 1,\ j = 1, 2, \ldots, q\}$ be the ordered set of fractional binary variables ranked by nonincreasing weights ρ_j. Now, the random selection of a variable y_k in Ψ can be performed in several ways, and we propose two alternatives. The first is based on the concept of restricted candidate list (RCL), often used in other GRASP studies: a threshold parameter $\alpha \in (0, 1)$ is assigned, and only those variables in Ψ such that $\rho_j \in [\rho^{\max} - \alpha(\rho^{\max} - \rho^{\min}), \rho^{\max}]$ are inserted into the RCL, where $\rho^{\min}$, $\rho^{\max}$ are respectively the minimum and the maximum values of ρ_j for $j \in \Psi$. Then, the variable is randomly selected from the RCL according to a uniform distribution. The second alternative does not consider a RCL but instead extracts the variable y_k according to a truncated geometric distribution over Ψ. More specifically, generate a value s according to the probability mass function

$$\Pr\{S = s\} = \frac{\varphi(1-\varphi)^{s-1}}{\theta}, \quad s = 1, 2, \ldots, |\Psi|, \tag{7}$$

$$\theta = 1 - (1-\varphi)^{|\Psi|}, \tag{8}$$

$$\varphi = 1 - \frac{\sum_{j\in\Psi} \rho_j}{|\Psi|\rho^{\max}}, \tag{9}$$

and take the sth variable in the ordered set Ψ. By this way the variables in Ψ have decreasing probabilities of being chosen. The choice of the parameter φ is self-explanatory: the more the maximum weight exceeds the average weight, the closer φ is to 1, so that the element of maximum weight is more likely to be selected.

After random selection of the fractional binary variable y_k, its value is forced to zero, and the modified continuous relaxation is solved again. There may arise two cases: if the relaxation has a feasible solution, then the step is repeated with the new relaxed solution. Otherwise, if no feasible solution exists, variable y_k is forced to one, and the continuous relaxation is solved again. If the variable y_k belongs to the set Q^1 then it is forced permanently to the value one. The entire step is still repeated. It is easy to verify that the construction phase will end up with a feasible solution $z^A = z(\mathbf{x}^A, \mathbf{y}^A)$ which is tentatively improved by means of a local search phase. To fully describe this latter, we just have to define the concept of neighborhood of a feasible solution **(x, y)** to MBP. A solution $(\mathbf{x}', \mathbf{y}')$ belongs to a neighbor of **(x, y)** if it is feasible and if it can be obtained by exchanging the value of two binary variables in **y**, say them y_r, y_s, as follows

$$y_r = 0, y'_r = 1, y_s = 1, y'_s = 0. \tag{10}$$

3 A bayesian stopping rule for GRASP

As we have seen in Sect. 2, a GRASP requires an appropriate stopping rule to be assigned for the repeated independent executions of the algorithm, determining how large the value of M should be. In practice, this issue has been disregarded in most studies, assuming a blanket rule that fixes in advance a large value of M. However, as noticed for the aforementioned examples, there are situations where devising an adaptive stopping rule can be highly beneficial to balance the computational effort and the quality of the approximate solution.

Let z^L and z^U be known lower and upper bounds to the optimum z^* of MBP. For instance, z^L can be obtained by solving the continuous relaxation of MBP, whereas z^U can be the value associated to any feasible solution. To simplify the derivation of the stopping rule, we make the reasonable assumption that the user can be satisfied by an approximate ψ-optimal solution having a relative error less than a specified tolerance $0 \le \psi \le 1$, that is

$$\frac{\bar{z} - z^*}{z^U - z^L} \le \psi. \tag{11}$$

Now apply the following transformation to the objective function of MBP: scale down each cost coefficient by the factor $\pi = (z^U - z^L)/\lceil z^U - z^L \rceil$ and subtract the constant $(z^L - 1)$. Moreover, round up the solution value generated by A to its integer ceiling, so that the transformed r.v. $\bar{Z}_T$ ranges in the set $\{1, 2, \ldots, r\}$, where $r = \lceil z^U - z^L \rceil + 1$ and subscript T is used to denote transformed values. It can be easily verified that a ψ-optimal solution $\bar{z}_T$ after the transformation corresponds to a ψ-optimal solution $\bar{z}$ for the original MBP. Indeed, if the inequality $(\bar{z}_T - z^*_T)/(r-1) < \psi$ holds, then

$$\frac{\bar{z} - z^*}{z^U - z^L} \leq \pi \frac{\bar{z}_T - z_T^*}{z^U - z^L} = \frac{\bar{z}_T - z_T^*}{\lceil z^U - z^L \rceil} = \frac{\bar{z}_T - z_T^*}{r - 1} < \psi. \tag{12}$$

In the following, subscript T will be dropped for transformed values. Denote by p_z and $F(z)$, respectively, the probability mass function (p.m.f.) and the cumulative distribution function (c.d.f.) of the r.v. $\bar{Z}$. It seems reasonable to stop the execution whenever the probability of improving the current best solution value $\bar{z}$, achieved during the first M independent executions of algorithm A, is below an assigned level of confidence ξ

$$F(\bar{z}) < \xi, \tag{13}$$

that is when $\bar{z} < z_\xi$, where z_ξ is the ξ-order quantile of the distribution $F(z)$. In order to test the validity of the stopping condition (13) a bayesian two decisions framework is developed, by letting d_0 be the action "accept hypothesis (13)" and d_1 the opposite action "reject hypothesis (13)". The sampling information is represented by a vector $t = (t_1, t_2, \ldots, t_r)$ whose ith component equals the number of elements of the sequence $\{z_j^A\}$, $j = 1, 2, \ldots, M$, falling into the ith value of the range for the r.v. $\bar{Z}$, that is $t_i = |\{j : z_j^A = i\}|$. The whole sample space is then composed by those r-vectors such that $t_i \geq 0$, $i = 1, 2, \ldots, r$, and $\sum_{i=1}^r t_i = M$. Since each run of the algorithm A is independent, the joint distribution of the r.v.'s t_i, $i = 1, 2, \ldots, r$, is multinomial

$$\Pr\{t_1 = n_1, t_2 = n_2, \ldots, t_r = n_r\} = \frac{M!}{n_1! n_2! \ldots n_r!} \prod_{i=1}^r p_i^{n_i}. \tag{14}$$

The space of parameters, denoted as Θ, corresponds to all p.m.f.'s defined over the set $i = 1, 2, \ldots, r$, that is to points in the $(r-1)$-simplex

$$S^{r-1} = \left\{ p = (p_1, p_2, \ldots, p_r) : p_i \geq 0, \sum_{i=1}^r p_i = 1 \right\}. \tag{15}$$

The choice of a suitable loss function should balance a trade-off between the quality of the approximation of the current $\bar{z}$ to the optimum z^* and the computational effort involved in performing further runs of A. The loss function considered here is

$$\begin{aligned} L(p, d_0) &= w_0 I_{(-\infty, \bar{z}]}(z_\xi), \\ L(p, d_1) &= w_1 I_{(\bar{z}, +\infty)}(z_\xi), \end{aligned} \tag{16}$$

where $I_E(\cdot)$ is the indicator function of the set E and w_0, w_1 are nonnegative real parameters, which can be interpreted as follows: w_0 is the penalty cost incurred when retaining the current best value $\bar{z}$ though a better approximation to z^* could have been attained with a probability greater than the confidence threshold ξ, whereas w_1 is the cost involved in performing a further iteration after the stopping condition being met.

The bayesian framework requires a prior probability measure $\tau(p)$ be assigned over the space of parameters Θ. A candidate prior $\tau(p)$ should ideally satisfy two properties: first, it should be general enough so that its support in Θ includes p.m.f.'s of different forms; second, although not strictly necessary, it is convenient that the posterior $\tau(p|t)$ given the sample t be manageable analytically, in order to reduce the computational effort.

The optimal bayesian decision is the one that minimizes the expected risk, defined as

$$R(\tau, d_l) = \int_\Theta L(p, d_l) \mathrm{d}\tau(p|t), \qquad l = 0, 1. \tag{17}$$

In particular, the expected risk corresponding to the loss functions (16) is given by

$$R(\tau, d_0) = \int_{\Theta} w_0 I_{(-\infty,\bar{z}]}(z_\xi) \mathrm{d}\tau(p|t) = w_0 \Pr\{z_\xi \leq \bar{z}|t\} = w_0 \Pr\{F(\bar{z}) \geq \xi|t\}, \quad (18)$$

$$R(\tau, d_1) = \int_{\Theta} w_1 I_{(\bar{z},+\infty)}(z_\xi) \mathrm{d}\tau(p|t) = w_1 \Pr\{\bar{z} < z_\xi|t\} = w_1 \Pr\{F(\bar{z}) < \xi|t\}. \quad (19)$$

Thus, the optimal bayesian decision is d_0 if the inequality

$$w_0 \Pr\{F(\bar{z}) \geq \xi|t\} < w_1 \Pr\{F(\bar{z}) < \xi|t\} \quad (20)$$

holds, and d_1 otherwise. It follows that the optimal decision is d_0 if

$$\Pr\{F(\bar{z}) < \xi|t\} > \frac{w_0}{w_0 + w_1} \quad (21)$$

holds, and d_1 if the converse is true. Consequently, in order to derive the optimal bayesian rule, the main efforts are directed to explicitly calculating the conditional probability in (21). In the sequel we will propose two different models of prior distribution $\tau(p)$ over Θ, deriving in both cases explicit optimal bayesian rules.

3.1 Dirichlet prior

The most natural choice for the prior $\tau(p)$ is the *Dirichlet distribution* over the $(r-1)$-dimensional simplex S^{r-1}, known as the conjugate prior for the parameters of a multinomial distribution; for an extensive treatment see (Wilks 1962). The r.v.'s $(p_1, p_2, \ldots, p_r)$ are said to follow a Dirichlet distribution over S^{r-1} if the joint $(r-1)$-dimensional density of $(p_1, p_2, \ldots, p_{r-1})$ is given by

$$f_{p_1,p_2,\ldots,p_{r-1}}(p_1, p_2, \ldots, p_{r-1}) = \frac{\Gamma\left(\sum_{i=1}^{r} \beta_i\right)}{\prod_{i=1}^{r} \Gamma(\beta_i)} \left(\prod_{i=1}^{r-1} p_i^{\beta_i - 1}\right) \left(1 - \sum_{i=1}^{r-1} p_i\right)^{\beta_r - 1}, \quad (22)$$

where $\Gamma(\alpha)$ is the gamma function $\int_0^\infty \lambda^{\alpha-1} e^{-\lambda} \mathrm{d}\lambda$ and β_i, $i = 1, 2, \ldots, r$, are positive real parameters, linked to the prior expected values of the p_i by the relationships

$$E[p_i] = \frac{\beta_i}{\sum_{i=1}^{r} \beta_i}, \quad i = 1, 2, \ldots, r. \quad (23)$$

Their values can be assigned on the basis of prior guesses about p_i, $i = 1, 2, \ldots, r$. If no hint is available, one can set $\beta_i = \beta$, $i = 1, 2, \ldots, r$, for some constant β, deriving a symmetric prior $\tau(p)$ over S^{r-1}. The particular choice $\beta = 1$ leads to the uniform distribution over S^{r-1}. As an alternative, one may consider the maximum entropy prior, with $\beta = 0$.

To describe in explicit form the stopping rule we need to evaluate the probability in (13), as accomplished in theorem 3.1 below. The two following properties of the Dirichlet distribution are needed:

Proposition 3.1 *If the r.v.'s $(p_1, p_2, \ldots, p_r)$ have prior Dirichlet distribution of parameters $(\beta_1, \beta_2, \ldots, \beta_r)$, then their posterior distribution, given the sample $t = (t_1, t_2, \ldots, t_r)$, is Dirichlet of parameters $(\beta_1 + t_1, \beta_2 + t_2, \ldots, \beta_r + t_r)$.*

Proposition 3.2 *If the r.v.'s $(p_1, p_2, \ldots, p_r)$ have a Dirichlet distribution of parameters $(\beta_1, \beta_2, \ldots, \beta_r)$, then the distribution of the r.v. $\sum_{i=1}^{\bar{z}} p_i$ is beta of parameters $\sum_{i=1}^{\bar{z}} \beta_i$ and $\sum_{i=\bar{z}+1}^{r} \beta_i$.*

Let $B(\mu, \nu) = B_1(\mu, \nu)$ be the beta function, and $I_u(\cdot, \cdot)$ the incomplete beta function

$$I_u(\mu, \nu) = \frac{B_u(\mu, \nu)}{B(\mu, \nu)} = \frac{1}{B(\mu, \nu)} \int_0^u \lambda^{\mu-1}(1-\lambda)^{\nu-1} \mathrm{d}\lambda. \tag{24}$$

We are now in a position to prove the

Theorem 3.1 *If the Dirichlet distribution over S^{r-1} is assumed, then*

$$Pr\{F(\bar{z}) < \xi | t\} = \frac{1}{B(\mu, \nu)} \int_0^\xi \lambda^{\mu-1}(1-\lambda)^{\nu-1} \mathrm{d}\lambda = I_\xi(\mu, \nu), \tag{25}$$

where $\mu = \sum_{i=1}^{\bar{z}} (\beta_i + t_i)$ and $\nu = \sum_{i=\bar{z}+1}^{r} (\beta_i + t_i)$.

Proof Relation (25) follows from

$$\Pr\{F(\bar{z}) < \xi | t\} = \Pr\left\{ \sum_{i=1}^{\bar{z}} p_i < \xi | t \right\}, \tag{26}$$

noticing that Propositions 3.1 and 3.2 together imply that the r.v. $\sum_{i=1}^{\bar{z}} p_i$, given the sample t, is beta distributed of parameters (μ, ν). □

Thus, for the Dirichlet distribution the stopping rule takes on a manageable form, as depicted by expression (25), and both requirements expressed above about the prior $\tau(p)$ are met. It can be noticed that $t_i = 0$ for $i = 1, 2, \ldots, \bar{z}$, and that $\sum_{i=\bar{z}+1}^{r} t_i = M$: this implies that the parameters μ, ν in (25) can actually be expressed as

$$\mu = \sum_{i=1}^{\bar{z}} \beta_i, \qquad \nu = \sum_{i=\bar{z}+1}^{r} \beta_i + M. \tag{27}$$

Consequently, the posterior distribution $\tau(p|t)$ gains information from the sample only through the total number M of observations falling into the set $\{\bar{z}+1, \bar{z}+2, \ldots, r\}$, but not through the specific values assumed by the observations within this set. This property makes the posterior distribution so easy to handle for the Dirichlet prior. However, the posterior is rather insensitive to the sampled values, and this is not a desirable feature of $\tau(p|t)$. Thus, in addition to the Dirichlet distribution, we propose a different prior which seems more suited to model the distribution of the sampled z_j^A, even at the expense of a narrower support over Θ.

3.2 Right-binomial prior

The second model of prior distribution $\tau(p)$ we introduce, referred to as *right-binomial* prior, is described by assigning an explicit formula for the unknown probabilities $p_i, i = 1, 2, \ldots, r$:

$$p_i = \binom{r-K}{r-i} U^{r-i}(1-U)^{i-K}, \tag{28}$$

(the coefficient $\binom{\alpha}{\beta}$ being zero whenever $\alpha < \beta$) which involves two independent r.v.'s K and U, satisfying the following assumptions:

- K is a discrete r.v. taking its values in the set $\{1, 2, \ldots, r\}$ with a binomial p.m.f.

$$f_K(k) = \binom{r-1}{k-1} \delta^{k-1}(1-\delta)^{r-k}, \quad k = 1, 2, \ldots, r, \tag{29}$$

  where $0 < \delta < 1$ is a given parameter;
- U is a continuous r.v. in the interval $(0, 1)$ with beta density

$$f_U(u) = \frac{1}{B(a,b)} u^{a-1}(1-u)^{b-1}, \tag{30}$$

  where $a, b > 0$ are given parameters.

Some remarks may guide a suitable choice of the parameters δ, a and b. The first $(K-1)$ components of the p.m.f. p_i are all equal to zero, while a binomial-like shape is displayed over the remaining right part of the set $\{1, 2, \ldots, r\}$. As a consequence, the r.v. K has an immediate interpretation: it represents the unknown optimum value. Moreover, it can be easily seen that

$$E[K] = (r-1)\delta + 1. \tag{31}$$

Thus, the parameter δ is linked to the prior guess about z^*, and relation (31) highlights the choice of its value. The lack of any belief about the value z^* can bring a pessimistic strategy into effect, by letting $E[K] = 1$. The values of the parameters a and b can be assessed by means of the prior guess about mean and variance of the r.v. Z^A. In fact, both $E[W]$ and $Var[W]$ can be easily evaluated and fixed according to the prior information; again, a pessimistic strategy can be pursued. This leads to a system of two equations in the two unknowns a and b.

We show how $E[Z^A]$ can be obtained; in a similar way the analogous explicit formula for the variance can be derived.

Theorem 3.2 *For the right binomial prior*

$$E[Z^A] = r - \frac{a}{a+b}[r - 1 - (r-1)\delta]. \tag{32}$$

Proof We have

$$\begin{aligned} E[Z^A] &= \sum_{k=1}^{r} f(k) \int_0^1 f(u) E[\bar{Z}|k,u]\mathrm{d}u \\ &= \sum_{k=1}^{r} f(k) \int_0^1 f(u) \left[\sum_{w=0}^{r-k} \binom{r-k}{w} u^w (1-u)^{r-k-w} (r-w) \right] \mathrm{d}u \\ &= \sum_{k=0}^{r-1} \binom{r-1}{k} \delta^k (1-\delta)^{r-1-k} \left[r - (r-k-1)\frac{a}{a+b} \right] \\ &= r - \frac{a}{a+b}[r - 1 - (r-1)\delta], \end{aligned} \tag{33}$$

and this proves the theorem. □

The following result expresses in explicit form the stopping rule for the right-binomial prior:

Theorem 3.3 *For the right-binomial prior*

$$\Pr\{F(\bar{z}) < \xi | t\} = \frac{\sum_{k=1}^{\bar{z}+1} \binom{r-1}{k-1} \delta^{k-1}(1-\delta)^{r-k} \left[\prod_{i=1}^{r} \binom{r-k}{r-i}^{t_i}\right] B_w(Mr-\Delta+a,\, \Delta-Mk+b)}{\sum_{k=1}^{r} \binom{r-1}{k-1} \delta^{k-1}(1-\delta)^{r-k} \left[\prod_{i=1}^{r} \binom{r-k}{r-1}^{t_i}\right] B(Mr-\Delta+a,\, \Delta-Mk+b)}, \tag{34}$$

where w is the unique value satisfying the relation $I_w(r-\bar{z}, \bar{z}-k+1) = \xi$ for $r-\bar{z} > 0$, $\bar{z}-k+1 > 0$, while $w = 1$ for $r-\bar{z} = 0$ or $\bar{z}-k+1 = 0$; also, $\Delta = \sum_{i=1}^{r} t_i i$, and it is assumed that $0! = 1, 0^0 = 1, 0^i = 0$ for $i > 0$.

Proof Conditioning on K and observing that $f(k|t) = 0$ for $k > \bar{z}+1$, one has

$$\begin{aligned} \Pr\{F(\bar{z}) < \xi | t\} &= \sum_{k=1}^{\bar{z}+1} f(k|t) \Pr\left\{ \sum_{i=r-\bar{z}}^{r-k} \binom{r-k}{i} U^i (1-U)^{r-k-i} < \xi | t, k \right\} \\ &= \sum_{k=1}^{\bar{z}} f(k|t) \Pr\{I_U(r-\bar{z}, \bar{z}-k+1) < \xi | t, k\} + f(\bar{z}+1|t). \end{aligned} \tag{35}$$

$I_v(r-\bar{z}, \bar{z}-k+1)$ is a function of v monotonically increasing over the interval (0, 1), and it assumes all values of the interval (0, 1), so that the equation $I_v(r-\bar{z}, \bar{z}-k+1) = \xi$ admits a unique solution w. As remarked, we take $w = 1$ when one of the arguments equals zero. Then (35) becomes

$$\sum_{k=1}^{\bar{z}+1} f(k|t) \Pr\{U < w | t, k\} = \sum_{k=1}^{\bar{z}+1} \frac{f(k,t)}{f(t)} \int_0^w \frac{f(t|k,u) f(k,u)}{f(k,t)} \mathrm{d}u. \tag{36}$$

Notice that $f(t|k,u)$ is a multinomial p.m.f.

$$f(t|k,u) = \frac{M!}{\prod_{i=1}^{r} t_i!} \prod_{i=1}^{r} \left[\binom{r-k}{r-i}^{t_i} u^{t_i(r-i)} (1-u)^{t_i(i-k)} \right]. \tag{37}$$

The value of $f(t)$ can be computed by conditioning on K and U

$$\begin{aligned} f(t) = \sum_{k=1}^{r} f(k) \int_0^1 f(u) f(t|k,u) \mathrm{d}u = \frac{M!}{\prod_{i=1}^{r} t_i!} \sum_{k=1}^{r} \binom{r-1}{k-1} \delta^{k-1}(1-\delta)^{r-k} \\ \times \left[\prod_{i=1}^{r} \binom{r-k}{r-i}^{t_i} \right] \frac{B(Mr-\Delta+a, \Delta-Mk+b)}{B(a,b)}. \end{aligned} \tag{38}$$

Putting (37) and (38) into (36) leads to (34). □

4 Computational tests

The proposed stopping rules have been implemented and tested using the two versions of GRASP described in Sect. 2 for solving instances of model FDVM. For the computational experiences, four difficult and time-consuming problems were considered, concerning the classification of the following benchmark datasets, publicly available from the UCI Machine Learning Repository (Hettich et al. 1998): "DNA", "led display"(Led), "satellite image" (Satellite) and "vehicle silhouette" (Vehicle). For these datasets, classification trees $FDSDT_{SLP}$, based on model FDVM and solved with a sequential algorithm (Orsenigo and Vercellis 2003), exhibits unsatisfactory performances in terms of accuracy and computational effort. We then used the two alternative versions of GRASP to solve FDVM at each node of the tree, in place of the sequential algorithm. To verify the effectiveness of the bayesian framework, we compared the classification trees generated by the proposed rules with those obtained by two simple stopping criteria, each performing a fixed number of executions of GRASP. These blanket rules, denoted as Bl_1 and Bl_2 in the computational tests, were forced to implement respectively 50 and 500 iterations of the randomized procedure (Table 1).

To assess the performance of the stopping rules, we evaluated the accuracy and the computing time required when the classification trees were generated by means of eight alternative algorithms. Four of these methods were derived by combining the blanket stopping rules with the way of selecting the fractional variable described in Sect. 2, according to the restricted list of candidates (RLC) or to a geometric selection (GEO); the remaining four algorithms were obtained by applying the bayesian decision for each type of randomization and for each form of the prior distribution; Bs_D stands for the Dirichlet distribution, whereas Bs_R denotes the right-binomial prior.

The computational experiences point out a number of interesting issues. The comparison with the methods based on the simple criteria shows that a significant computational saving can be achieved with the proposed stopping rule, whether the Dirichlet prior or the right-binomial prior is assumed as the prior distribution, without significantly compromising the overall accuracy. Actually, the correctness of the classification is comparable to the best values obtained by the most time-consuming blanket rule. Notice also that the bayesian stopping rule permits to achieve high accuracies either when the fractional variable is randomly selected from the restricted candidate list or it is chosen according to the truncated geometric distribution. Furthermore, there is a mild dependence from the specific form of the prior distribution adopted, with a slight preference for the right-binomial prior. As might be expected, for the blanket rules the accuracy of the classification is preserved only by the criteria involving the largest number of iterations.

Table 1 Computational tests on benchmark datasets: accuracy results (%) and computing times (mm:ss)

Dataset	Method								
		Bl_1		Bl_2		Bs_D		Bs_R	
	$FDSDT_{SLP}$	RLC	GEO	RLC	GEO	RLC	GEO	RLC	GEO
DNA	85.2	86.4	87.1	92.1	92.1	89.4	88.6	91.5	90.5
3 classes	02:12	04:36	04:39	46:13	47:16	14:26	14:27	14:25	14:24
Led	63.4	65.6	64.9	71.8	70.2	69.2	69.8	71.3	71
10 classes	04:48	09:10	09:12	93:22	93:26	35:04	33:06	35:06	35:08
Satellite	75.3	73.2	73.7	82.2	81.8	79.3	79.7	80.9	81.8
6 classes	03:24	05:30	05:31	55:48	55:50	17:11	17:14	17:10	17:09
Vehicle	76.6	77	77.4	83.4	83.6	81.2	81.5	82.1	81.3
4 classes	02:30	04:27	04:29	45:07	45:10	14:13	14:15	14:12	14:14

References

Betrò, B., Schoen, F.: Sequential stopping rules for the multistart algorithm in global optimisation. Math. Program. **38**, 271–286 (1987)
Betrò, B., Schoen, F.: Optimal and suboptimal stopping rules for the multistart algorithm in global optimisation. Math. Program. **57**, 445–458 (1992)
Betrò, B., Vercellis, C.: Bayesian nonparametric inference and Monte Carlo optimization. Optimization **17**, 681–694 (1986)
Boender, C., Rinnooy Kan, A.: Bayesian stopping rules for multistart global optimization methods. Math. Program. **37**, 59–80 (1987)
Boender, C., Rinnooy Kan, A., Vercellis, C.: Stochastic Optimization Methods. pp. 94–112. World Scientific (1987)
Canuto, S., Resende, M., Ribeiro, C.: Local search with perturbations for the prize-collecting steiner tree problem in graphs. Networks **38**, 50–58 (2001)
Feo, T., Resende, M.: Greedy randomized adaptative search procedures. J. Global Optimiz. **6**, 109–133 (1995)
Festa, P., Resende, M.: GRASP: An Annotated Bibliography. pp. 325–367. Kluwer Academic Publishers (2002)
Fumero, F., Vercellis, C.: Capacity management through lagrangean relaxation: an application to tires production. Prod. Plan. Control **7**, 604–614 (1996)
Fumero, F., Vercellis, C.: Integrating distribution, lot-sizing and machine loading via lagrangean relaxation. Int. J. Prod. Econ. **49**, 45–54 (1997)
Hart, W.: Sequential stopping rules for random optimization methods with applications to multistart local search. SIAM J. Optimiz. **9**, 270–290 (1999)
Hettich, S., Blake C., Merz, C.: UCI repository of machine learning databases. (1998). URL http://www.ics.uci.edu/∼mlearn/MLRepository.html
Orsenigo, C., Vercellis, C.: Multivariate classification trees based on minimum features discrete support vector machines. IMA J. Manage. Math. **14**, 221–234 (2003)
Orsenigo, C., Vercellis, C.: Discrete support vector decision trees via tabu-search. J. Comput. Stat. Data Anal. **47**, 311–322 (2004a)
Orsenigo, C., Vercellis, C.: One-against-all multicategory classification via discrete support vector machines. In: Ebecken N. et al. (eds.) Data Mining IV. pp. 255–264. WIT Press (2004b)
Prais, M., Ribeiro, C.: Reactive grasp: An application to a matrix decomposition problem in TDMA traffic assignment. INFORMS J. Comput. **12**, 164–176 (2000)
Resende, M., Ribeiro, C.: Greedy Randomized Adaptive Search Procedures. pp. 219–249. Kluwer Academic Publishers (2003)
Wilks, S.: Mathematical Statistics. Wiley (1962)