Matheuristics: Optimization, Simulation and Control

Marco A. Boschetti[1], Vittorio Maniezzo[1], Matteo Roffilli[1], and Antonio Bolufé Röhler[2]

[1] University of Bologna, Bologna, Italy
[2] University of Habana, Habana, Cuba

1 Extended Abstract

Matheuristics are heuristic algorithms made by the interoperation of metaheuristics and mathematic programming (MP) techniques. An essential feature is the exploitation in some part of the algorithms of features derived from the mathematical model of the problems of interest, thus the definition "model-based metaheuristics" appearing in the title of some events of the conference series dedicated to matheuristics [1]. The topic has attracted the interest of a community of researchers, and this led to the publication of dedicated volumes and journal special issues, [13], [14], besides to dedicated tracks and sessions on wider scope conferences.

The increasing maturity of the area permits to outline some trends and possibilities offered by matheuristic approaches. A word of caution is needed before delving into the subject, because obviously the use of MP for solving optimization problems, albeit in a heuristic way, is much older and much more widespread than matheuristics. However, this is not the case for metaheuristics, and also the very idea of designing MP methods specifically for heuristic solution has innovative traits, when opposed to exact methods which turn into heuristics when enough computational resources are not available.

1.1 Optimization

Some approaches using MP combined with metaheuristics have begun to appear regularly in the matheuristics literature. This combination can go two-ways, either using MP to improve or design metaheuristics or using metaheuristics for improving known MP techniques, even though the first of these two directions is by far more studied.

When using MP embedded into metaheuristics, the main possibility appears to be improving local search (see [8] for a detailed overview). A seminal work in this direction is local branching [9], where MP is used to define a suitable neighborhood to be explored exactly by a MIP solver. Essentially, only a number of decision variables is left free and the neighborhood is composed by all possible value combination of these free variables.

The idea of an exact exploration of a possibly exponential size neighborhhood is at the heart of several other approaches. One of the best known is possibly Very

M.J. Blesa et al. (Eds.): HM 2009, LNCS 5818, pp. 171–177, 2009.

Large Neighborhood Search (VLNS, [3]). This method can be applied when it is possible to define the neighborhood exploration as a combinatorial optimization problem itself. In this case it could be possible to solve it efficiently, and it becomes possible the full exploration of exponential neighborhoods.

Complementary to this last is the *corridor method* [15], where a would-be large exponential neighborhood is kept of manageable size by adding exogenous constraint to the problem formulation, so that the feasible region is reduced to a "corridor" around the current solution.

Several other methods build around the idea of solving via MP the neighborhood exploration problem, they differ in the way the neighborhood is defined. For example, an unconventional way of defining it is proposed in the *dynasearch* method [7], where the neighborhood is defined by the *series of moves* which can be performed at each iteration, and dynamic programming is used to find the best sequence of simple moves to use at each iteration.

However, MP contributed to metaheuristics also along two other opposite lines: improving the effectiveness of well-established metaheuristics and providing the structural basis for designing new metaheuristics.

As for the first line, MP hybrids are reported for most known metaheuristics: tabu search, variable neighborhood search, ant colony optimization, simulated annealing, genetic algorithms, scatter search, etc. Particularly appealing appear to be genetic algorithms, for which a number of different proposals were published, with special reference to how to optimize the crossover operator. For example, Yagiura and Ibaraki [17] propose to fix the solution parts common to both parents and to optimize only on the remaining variables, while Aggarwal et al. [2] make a set consisting of the union of the components of the parent solutions and optimize within that set.

As for the second line, the proposals are different, but they still have to settle and show how they compare on a broader range of problems, other than those for which they were originally presented. One example is the so-called *Forward and Backward* (*F&B*) approach [4] which implements a memory-based look ahead strategy based on the past search history. The method iterates a partial exploration of the solution space by generating a sequence of enumerative trees of two types, called forward and backward trees, such that a partial solution of the forward tree has a bound on its completion cost derived from partial solutions of the backward tree, and vice-versa.

Another example uses classical decomposition strategies, namely Langrangean, Benders and Dantzig-Wolfe, to obtain metaheuristic algorithms, with a characteristic feature of evolving both a feasible solution and a bound to its cost [6].

1.2 Simulation

Simulation "is an attempt to model a real-life or hypothetical situation on a computer so that it can be studied to see how the system works. By changing variables, predictions may be made about the behaviour of the system" (source: Wikipedia). It is closely related to optimization, both because they often target the same applications in decision-making contexts and because prediction can often be made by optimizing a mathematical model.

It is with reference to this last possibility that matheuristics have been used in a simulation context. Optimization was used to determine how will system variables distribute under the assumption that the whole system leans toward a minimum free energy status.

For example, an application to traffic flow simulation was reported in [10]. In this case, the optimality assumption was formalized by the Wardrop's principles [16], which were taken as a guide for designing the objective function to be optimized by the matheuristic. The problem to solve could be stated as: given a road network with all needed parameters attached to the roads, an origin-destination matrix, and possibly traffic counters on some roads, determine the traffic flows on the roads. This can be formalized as a min cost multicommodity network flow (MCMNF) problem, with a nonlinear objective function, since the time for a driver to travel a road is a nonlinear function of the level of congestion of the road itself, as dictated by queuing theory. The cited application simulated the steady state in hours of maximum congestion and was applied to case studies relative both to cities (see figures 1 and 2 for an application to the case of La Habana, Cuba) and to whole regions, where the OD matrix counted thousands of rows and columns and the road network was composed of hundreds of thousands of arcs. In all cases the simulation time could be limited to a few minutes on a standard PC.

Figure 1 shows how congestion is correctly handled and, when paths followed by different drivers in congested situations are plotted, one can see that different choices were made for going from one same origin to one same destination. This ensures that no faster alternative way is available (Wardrop's first principle). Figure 2 shows the overall flow distribution for the 12 zones we considered in this study.

1.3 Process Control

Process control is "the task of planning and regulating a process, with the objective of performing it in an efficient, effective and consistent manner" (source:

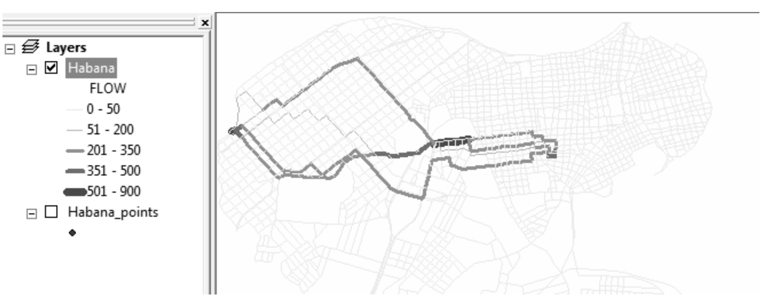

Fig. 1. Paths in Habana

Fig. 2. Flows in Habana

WikiPedia). Again, the request for efficiency and effectiveness calls for optimization, thus matheuristics can contribute to some applications.

Different use cases were reported. One refers to the identification of the best time-dependent bandwidth allocation parameters for peer-to-peer (p2p) nodes which should upload and download data [11], [5]. In this application, the proposed algorithm was a Lagrangean metaheuristics which could be fully distributed on the nodes of the peers themselves, allowing each node to optimize its own parameter in such a way that a global optimum would eventually emerge.

Figure 3 shows the simulated trajectory of the global network throughput, when nodes and connections were dynamically added or removed from the network. It can be seen how, after a brief startup where the nodes still had to decide who to connect with, a globally optimized state is quickly reached and fast adaptations to network structural variations could be ensured providing a

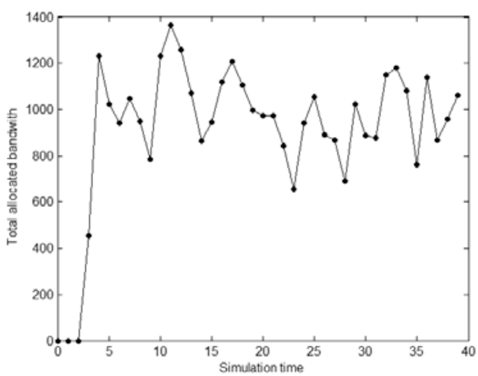

Fig. 3. Total system throughput

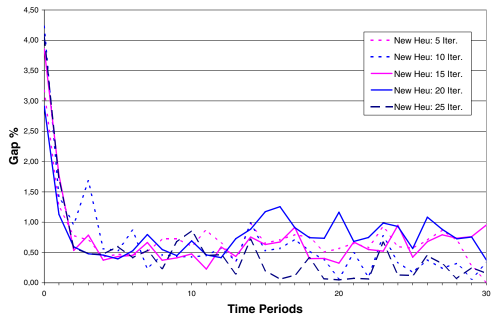

Fig. 4. Gap between the LP-relaxation and the heuristic solution

near optimal total system throughput. Figure 4 shows the evolution of the percentage gap between the LP-relaxation and the solution value provided by the Lagrangean metaheuristic for different parameter values. After the startup the gap is within the 1%.

Another application of matheuristics to process control aimed at determining the best distribution policy for a region-wide water network [12]. In that application, different infrastructural expansion of a current water supply network had to be compared in front of the needs and requests of different stakeholders, such as farmers, industries, civil population, water management authority, etc. A number of alternatives were identified and, for each of them, an optimal distribution policy had to be determined in order to assess the effectiveness of the particular infrastructure for each interest group. The distribution policies had to be able to cope with different environmental situations, such as years of drought followed by years of heavy rain.

The solution proposed made use of a *Receding Horizon Heuristic* (RHH), which is the metaheuristic counterpart of a well known process control technique named Model Predictive Control (MPC), or Receding Horizon Control (RHC). RHC is based on an iterative, finite horizon optimization of a model, and such is also the derived (meta)heuristic. At time t, the current system state is sampled and a cost minimizing control strategy is computed for a relatively short time horizon in the future: $[t, t+T]$. Only the first step of the control strategy is implemented, that relative to time t, then the system state is sampled again and the calculations are repeated starting from time $t+1$. This yields a new predicted state path which is implemented only in its first interval, that relative to time $t+1$. Notice how the planning horizon keeps being shifted forward, hence the names RHC and RHH.

Computational results for this use case are plotted in figures 5 and 6. These show the level of service which could be guaranteed to the major town and the level of the biggest lake in the area, in a simulation considering 10 years of very different raining conditions (drought, heavy rain, than a run of drought years before returning to normality).

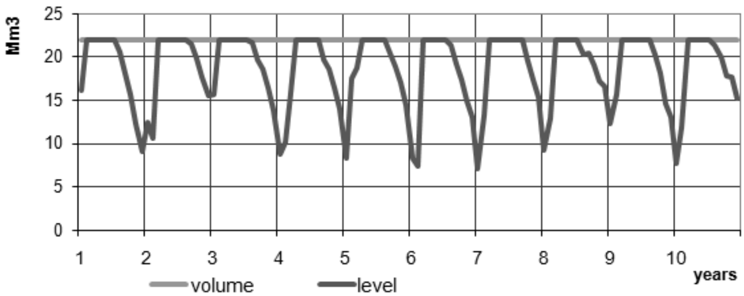

Fig. 5. Level of the reservoir

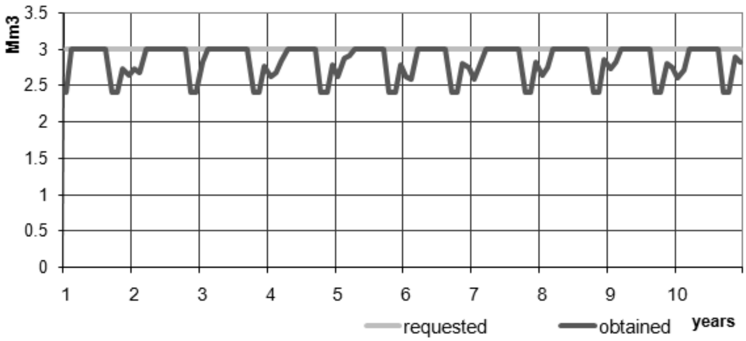

Fig. 6. City water requests

Figure 5 shows how the level of the reservoir, when compared with its usable capacity, decreases significantly during the summers, to recover during the winters. Figure 6 shows how the level of service to the main city, essentially making reference to civil usage such as drinking or washing, does never fall under an acceptable 80% of the request, which in turns will cause severe shortage for agricultural usage in drought years.

References

1. Matheuristics conferences page (2009), `http://astarte.csr.unibo.it/matheuristics`
2. Aggarwal, C., Orlin, J., Tai, R.: An optimized crossover for the maximum independent set. Operations Research 45, 226–234 (1997)
3. Ahuja, R., Ergun, O., Orlin, J., Punnen, A.: A survey of very large-scale neighborhood search techniques. Discrete Applied Mathematics 123(1-3), 75–102 (2002)
4. Bartolini, E., Mingozzi, A.: Algorithms for the non-bifurcated network design problem. J. of Heuristics 15(3), 259–281 (2009)
5. Boschetti, M., Jelasity, M., Maniezzo, V.: A fully distributed lagrangean metaheuristic for a p2p overlay network design problem. In: Hartl, R. (ed.) Proceedings MIC 2005 (2005)

6. Boschetti, M., Maniezzo, V.: Benders decomposition, lagrangean relaxation and metaheuristic design. J. of Heuristics 15(3), 283–312 (2009)
7. Congram, R., Potts, C., Van de Velde, S.: An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem. INFORMS Journal on Computing 14(1), 52–67 (2002)
8. Dumitrescu, I., Stützle, T.: Usages of exact algorithms to enhance stochastic local search algorithms. In: Maniezzo, V., Stützle, T., Voss, S. (eds.) Matheuristics: Hybridizing metaheuristics and mathematical programming, OR/CS Interfaces Series. Springer, Heidelberg (2009)
9. Fischetti, M., Lodi, A.: Local branching. Mathematical Programming B 98, 23–47 (2003)
10. Gabrielli, R., Guidazzi, A., Boschetti, M.A., Roffilli, M., Maniezzo, V.: Practical origin-destination traffic flow estimation. In: Proc. Third International Workshop on Freight Transportation and Logistics (Odysseus 2006), Altea, Spain (2006)
11. Maniezzo, V., Boschetti, M., Jelasity, M.: An ant approach to membership overlay design. In: Dorigo, M., Birattari, M., Blum, C., Gambardella, L.M., Mondada, F., Stützle, T. (eds.) ANTS 2004. LNCS, vol. 3172, pp. 37–48. Springer, Heidelberg (2004)
12. Maniezzo, V., Boschetti, M., Roffilli, M.: Matheuristics in simulation: a case study in water supply management. In: Caserta, M., Voss, S. (eds.) Proceedings MIC 2009, VIII Metaheuristic International Conference (2009)
13. Maniezzo, V., Stützle, T., Voss, S. (eds.): Matheuristics: Hybridizing Metaheuristics and Mathematical Programming. Annals of Information Systems, vol. 10. Springer, Heidelberg (2009)
14. Maniezzo, V., Voss, S., Hansen, P.: Special issue on mathematical contributions to metaheuristics editorial. Journal of Heuristics 15(3) (2009)
15. Sniedovich, M., Voss, S.: The corridor method: a dynamic programming inspired metaheuristic. Control and Cybernetics 35(3), 551–578 (2006)
16. Wardrop, J.G.: Some theoretical aspects of road traffic research, volume PART II, vol. 1. Institute of Civil Engineers, Palo Alto (1952)
17. Yagiura, M., Ibaraki, T.: The use of dynamic programming in genetic algorithms for permutation problems. European Journal of Operational Research 92, 387–401 (1996)