



HAL
open science

Model Enhancement: Improving Theoretical Optimization with Simulation

Bruno Bachelet, Loïc Yon

► **To cite this version:**

Bruno Bachelet, Loïc Yon. Model Enhancement: Improving Theoretical Optimization with Simulation. *Simulation Modelling Practice and Theory*, 2007, 15 (6), pp.703-715. 10.1016/j.simpat.2007.02.003 . hal-01703303

HAL Id: hal-01703303

<https://hal.science/hal-01703303>

Submitted on 7 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**Model Enhancement: Improving Theoretical
Optimization with Simulation**

Bruno Bachelet¹ and Loïc Yon²
LIMOS, UMR 6158-CNRS,
Université Blaise-Pascal, BP 10125, 63173 Aubière, France.

Research Report LIMOS/RR05-03

¹bruno.bachelet@isima.fr - <http://frog.isima.fr/bruno>

²loic.yon@isima.fr - <http://frog.isima.fr/loic>

Abstract

When using optimization techniques based on mathematical models, we often need to make important simplifications. The solution thus provided, even if proven to be theoretically one of the best, might not be so good in practice. Simulation can be used to evaluate the actual performance of the solution. We propose here a coupling between optimization and simulation that tries to improve the solution provided by a mathematical model. This approach, named "model enhancement" here, still focuses on optimizing the theoretical objective function, contrary to the common optimization-simulation coupling that focuses on improving the objective function evaluated from simulation. We propose to illustrate this approach on a routing problem, and present numerical results on the quality of the solution and the efficiency of both coupling approaches.

Keywords: optimization, discrete-event simulation, simulation optimization, routing problem.

Résumé

Lorsqu'on utilise des techniques d'optimisation qui reposent sur des modèles mathématiques, on est souvent obligé de faire d'importantes simplifications. La solution ainsi produite, même s'il est théoriquement prouvé qu'elle est l'une des meilleures, peut ne pas être si bonne que cela en pratique. La simulation peut être employée pour évaluer la performance véritable de la solution. Nous proposons ici un couplage entre optimisation et simulation qui tente d'améliorer la solution produite par un modèle mathématique. Cette approche, nommée "enrichissement de modèle" ici, se concentre toujours sur l'optimisation de la fonction objectif théorique, contrairement au couplage optimisation-simulation classique qui se concentre sur l'amélioration de la fonction objectif évaluée par simulation. Nous proposons d'illustrer cette approche sur un problème de routage, et présentons des résultats numériques sur la qualité de la solution et l'efficacité des deux approches de couplage.

Mots clés : optimisation, simulation à événements discrets, couplage optimisation-simulation, problème de routage.

Acknowledgements / Remerciements

We would like to thank Antoine Mahul, PhD Student at Blaise Pascal University, Clermont-Ferrand, France, for his invaluable advice to formalize model enhancement.

Abstract

When using optimization techniques based on mathematical models, we often need to make important simplifications. The solution thus provided, even if proven to be theoretically one of the best, might not be so good in practice. Simulation can be used to evaluate the actual performance of the solution. We propose here a coupling between optimization and simulation that tries to improve the solution provided by a mathematical model. This approach still focuses on optimizing the theoretical objective function, contrary to the common optimization-simulation coupling that focuses on improving the objective function evaluated from simulation. We propose to illustrate this approach on a routing problem, and present numerical results on the quality of the solution and the efficiency of both coupling approaches.

Introduction

The goal of this paper is to discuss a way to improve the practical quality of a solution provided by an optimization process. There are several advanced optimization techniques (mixed integer programming: branch-and-bound, branch-and-cut... [7], decomposition methods: Benders, Dantzig-Wolfe... [5]) that can solve efficiently problems formalized with mathematical models. Major results have been stated to prove the optimality or the quality of the solution (approximation algorithms can ensure the solution found to be close to the optimal solution according to a given precision [4]), and to ensure the efficiency of the techniques (their complexity, their speed to converge to a solution...).

However, these methods have significant drawbacks when looking for their practical implementation. First, they are not robust to changes in the structure of the problem: adding a new kind of constraints might make the problem unsolvable with the previous optimization technique (e.g. linear constraints, solved with the simplex method, that become non-linear). Secondly, and more of concern in this paper, major simplifications on the modeling of the problem often have to be considered. As a result, a solution that is optimal in theory may not be so good in practice.

Therefore, we propose an optimization-simulation coupling called *model enhancement*, that attempts to reinforce the mathematical model in order to make the solution more adapted to practice than a straightforward optimization. Section 1 recalls the common optimization-simulation coupling sketch, usually called *simulation optimization*, and formalizes the problem that we propose to discuss in this article. Section 2 explains what straightforward optimization implies on the formulation and the solutions of the problem. Section 3 finally presents the idea of model enhancement, its goal and sketch. This proposition is illustrated in Section 4 through a routing problem. The approaches presented in the previous sections are implemented for this problem, and the quality of their solution and their computational efficiency are compared.

1 Simulation Optimization

An optimization problem can be expressed as finding the best solution x to a real problem (P_r) , i.e. minimizing or maximizing a function $f_r(x)$. A solution x fits problem (P_r) if it satisfies a set of constraints that defines the space C_r of feasible solutions.

$$(P_r) \begin{cases} \text{optimize} & f_r(x) \\ \text{subject to} & x \in C_r \end{cases}$$

During the modeling phase, the real problem (P_r) must be approximated. The common formulation of simulation optimization is expressed by problem (P_s). The fact that a function (resp. a space of solutions) a is approximated by another function (resp. another space of solutions) b is denoted $b \sim a$.

$$(P_s) \left\{ \begin{array}{l} \text{optimize} \quad f_s(x) = g(x, \lambda), \quad f_s \sim f_r \\ \text{subject to} \quad \left. \begin{array}{l} x \in C_s \\ \lambda \in \Lambda_s(x) \end{array} \right\} \sim C_r \end{array} \right.$$

C_s represents constraints on a solution x . It usually defines the basic structure of a feasible solution (for instance, the solution must be a cycle in a graph). λ is a vector of measures that is returned by the simulation for a solution x (cf. Figure 1).

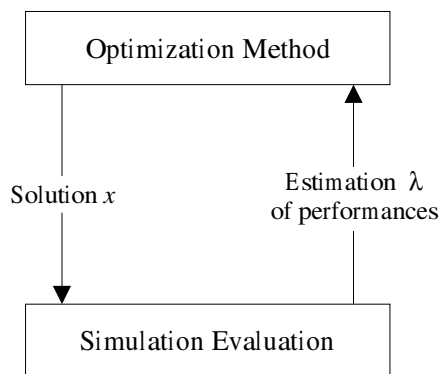


Figure 1: Simulation optimization sketch.

$\Lambda_s(x)$ is the set of feasible solutions for λ , according to a given solution x . It is defined by implicit constraints of the simulation model (given a solution x , the simulation returns the estimation λ), and by explicit constraints on some performance measures (e.g. estimated delay must not exceed a given limit, or simply the solution is acceptable). That means $\Lambda_s(x)$ contains either one solution (x is feasible based on the simulation evaluation), or no solution (x is not feasible based on the simulation evaluation).

The objective function f_r of the real problem is approximated by f_s . We propose to parameterize this function on vector λ : $f_s(x) = g(x, \lambda)$. For instance, λ can represent estimated delays for solution x , and $g(x, \lambda)$ an estimation of the quality of service of solution x according to these delays.

Simulation optimization explores the set of solutions C_s to optimize f_s . For each solution x , the simulation estimates vector λ . If λ satisfies all the constraints in $\Lambda_s(x)$, the solution x is accepted, and the objective function $g(x, \lambda)$ is evaluated.

Several methods are proposed to solve problem (P_s). A classification in four major approaches can be found in [6] and [1]: gradient based search, stochastic approximation, response surface and heuristic search. These methods are robust to changes in the objective function or in the constraints of the problem. However, they only represent a few optimization techniques and their efficiency and convergence are not always ensured.

2 Straightforward Optimization

Optimization techniques, independent of simulation, usually need a mathematical model. This representation is also an approximation of the real problem (P_r).

$$(P_o) \begin{cases} \text{optimize} & f_o(x), f_o \sim f_r \\ \text{subject to} & x \in C_o, C_o \sim C_r \end{cases}$$

However, we can reasonably assume that the simulation optimization problem (P_s) is closer to the real problem than the straightforward optimization problem (P_o). For the purpose of model enhancement, we now make some assumptions on the structure of (P_o).

$$(P_o) \begin{cases} \text{optimize} & f_o(x) = g(x, \lambda) \\ \text{subject to} & x \in C_o \supset C_s \\ & \lambda \in \Lambda_o(x), \Lambda_o \sim \Lambda_s \end{cases}$$

- The constraints C_s describe the basic structure of a feasible solution. Hence, we can reasonably assume that the constraints defining C_o are approximations, and even relaxations, of the constraints defining C_s . It means that $C_o \supset C_s$.
- We consider that the objective functions f_o and f_s are identical: $f_o(x) = g(x, \lambda)$. However, we assume that the space of solutions Λ_o is an approximation of the space of solutions Λ_s , which implicitly makes f_o an approximation of f_s .

Depending on the structure of the problem, various optimization techniques can be considered to solve (P_o). However, once a particular method m has been chosen to solve the problem, it is difficult to deal with changes on the kind of constraints of the problem. For model enhancement purpose, we consider changes only on the constraints that describe Λ_o . Let us denote K_m the set of all the families of constraints that can be managed by method m . That means problem (P_o) is solvable by method m only if $\Lambda_o \in K_m$.

3 Model Enhancement

From the previous assumptions, if $\Lambda_o(x)$ is never empty, all solutions x in C_o are feasible solutions of problem (P_o). As $C_o \supset C_s$, any solution of the simulation optimization problem (P_s) is a feasible solution of (P_o). In particular, any optimal solution x_s^* of (P_s) is a feasible solution of (P_o). The idea of model enhancement is to find the family of constraints $\Lambda_o \in K_m$ such that the optimal solution x_o^* of (P_o) is one of the optimal solutions x_s^* of (P_s). We can state the model enhancement problem (P_e) as follows.

$$(P_e) \begin{cases} \text{optimize} & f_s(x_o^*) = g(x_o^*, \lambda_s) \\ \text{subject to} & \Lambda_o \in K_m, \Lambda_o \sim \Lambda_s \\ & (x_o^*; \lambda_o^*) \text{ optimal solution of } (P_o) \\ & \lambda_s \in \Lambda_s(x_o^*) \end{cases}$$

(P_e) is a very hard problem. However, one way to find a good solution to the problem may be to approximate Λ_s by Λ_o as precisely as possible. Thus, λ_s and λ_o^* will have similar values and the theoretical evaluation $g(x_o^*, \lambda_o^*)$ of the optimal solution x_o^* of problem (P_o) will be close to its simulation evaluation $g(x_o^*, \lambda_s)$.

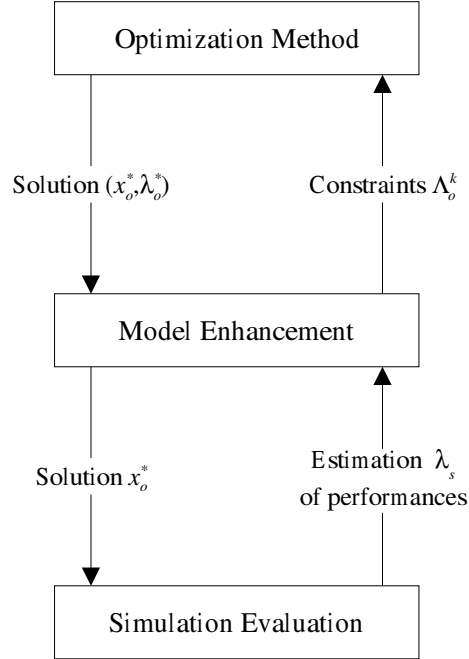


Figure 2: Model enhancement heuristic.

Algorithm 1 and Figure 2 describe a heuristic approach for model enhancement that iteratively modifies Λ_o , based on the simulation evaluation λ_s of an optimal solution of (P_o) .

Algorithm 1 *Model enhancement heuristic.*

```

k ← 0;
let n be the maximal number of iterations;
let  $\Lambda_o^k$  be an approximation of  $\Lambda_s$ ;

repeat
  let  $(x_o^*, \lambda_o^*)$  be an optimal solution of  $(P_o)$  with  $\Lambda_o = \Lambda_o^k$ ;
  let  $\lambda_s$  be the simulation evaluation of  $x_o^*$ ;
   $\Lambda_o^{k+1} \leftarrow h(\Lambda_o^k, \lambda_s)$ ;
  k ← k + 1;
until  $|g(x_o^*, \lambda_o^*) - g(x_o^*, \lambda_s)| < \varepsilon$  or  $k \geq n$ ;
  
```

Function h represents the way to modify Λ_o at each iteration. It needs to be defined more precisely for any specific problem. In the next section, we propose to illustrate this heuristic on a bus routing problem.

4 Study of a Routing Problem

We propose now to study a bus routing problem in order to illustrate the discussion of the previous sections. We detail the simulation optimization method, a straightforward optimization approach and the model enhancement heuristic for this problem. We also present a practical comparison of these three approaches.

4.1 Problem Presentation

We focus on the following problem: let us consider a public transportation company in an urban network, such as a bus company. Basically, this company needs to design low cost bus routes while satisfying potential customers.

Let us consider a directed graph $G = (V, E)$ where V is the set of vertexes and E the set of edges. Each vertex represents a potential bus stop or a crossroad in the real-life network. Each edge represents a road between two stops or crossroads.

We assume the customer demands are known, i.e. there is a description of the moves the customers need to perform in the urban network. A customer demand $d \in D$ is defined by a tuple $(o_d; s_d; Q_d; t_d^{min}; t_d^{max})$, where $o_d \in V$ is the origin of the move, $s_d \in V$ the destination, Q_d the throughput of customers for this demand. t_d^{min} and t_d^{max} are reference times for the demand: t_d^{min} is the time it takes to a vehicle (such as a bus) to move from o_d to s_d ; and t_d^{max} is the time of travel of a pedestrian from o_d to s_d .

In this article, we limit the problem to find a transportation system Γ that meets the following requirements:

- Γ is a set of directed cycles in graph G (for clarity reasons, we consider further only one bus cycle);
- the length of the bus route, i.e. the time it takes to the bus to move along the route, must be less than a given threshold T ;
- Γ should maximize the satisfaction of the customers.

Let us denote t_d the time it takes to a customer with demand d to travel through the network. His satisfaction can be represented by a function $\phi_d(t_d)$ defined as follows: if $t_d \leq 2t_d^{min}$ then the customer is fully satisfied and $\phi_d(t_d) = 1$; if $t_d \geq 2t_d^{max}$ then the customer is not satisfied at all and $\phi_d(t_d) = 0$; and between these two limits, the closer to the minimum, the better the customer is satisfied. Figure 3 illustrates the satisfaction function.

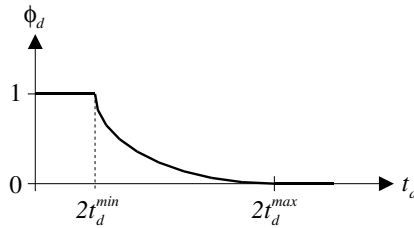


Figure 3: Satisfaction function for demand d .

Finally, we define w_i the time customers are waiting at a bus stop i . Back to our notations of the previous sections, λ is the vector of all the t_d and the w_i values. λ_o will be their theoretical estimation (from optimization) and λ_s their practical estimation (from simulation).

Let us denote x a solution to the problem, i.e. $x_e = 1$ if edge e is part of the bus route, and $x_e = 0$ otherwise. The objective is to maximize the satisfaction of the customers, i.e. function $g(x, \lambda) = \sum_{d \in D} Q_d \Phi_d(t_d)$.

4.2 Simulation Optimization

We propose to solve the problem using the tabu search metaheuristic in the simulation optimization sketch (P_s). The tabu search was first introduced by [2, 3]. It searches for a good solution x in the space of solutions C_s . C_s represents bus routes with their length below T . Moreover, we force the structure of the bus routes in C_s to be a *geodesic* (the search is thus facilitated). A geodesic is a directed cycle defined by a given number of points called *control points*. Each control point is connected to its successor in the cycle by a shortest path (cf. Figure 4).

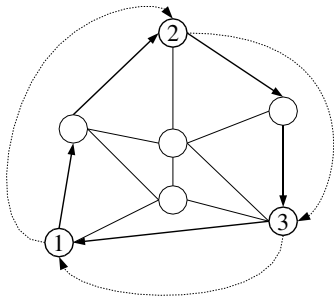


Figure 4: An example of 3-points geodesic.

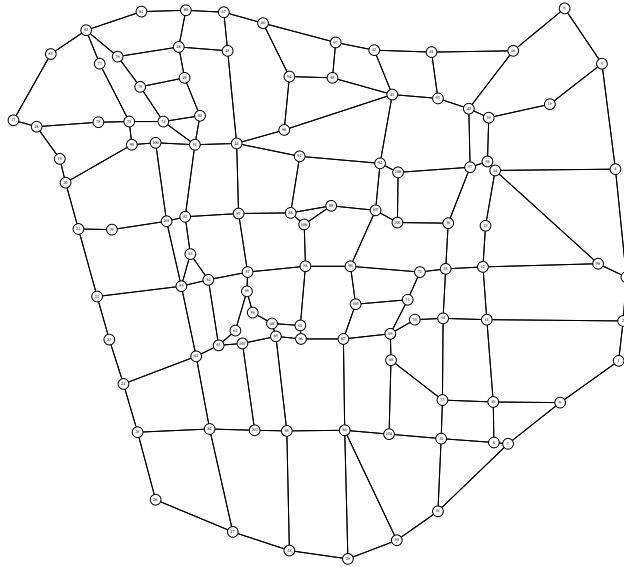


Figure 5: Graph modeling Clermont-Ferrand (France) downtown.

Neighborhood structures for the heuristic rely on moving, adding or deleting a control point. The tabu list contains the control points that have been modified in the last few iterations. We have implemented a two-level heuristic using aspiration criteria (accepting a tabu move under some conditions) and diversification strategy (exploring another area of the space of solutions).

For a given solution x of the simulation optimization problem (P_s), simulation estimates the time of travel t_d of the customers with demand d and their waiting time w_i at bus stop i (reminder: all these values are stored in vector λ). Besides the implicit constraints of simulation, we consider no additional constraints in Λ_s that can discard a solution x based on its estimation λ . That means any solution $x \in C_s$ is a feasible solution of (P_s) (it is a necessary condition for model enhancement to work).

We consider now a graph with 109 vertexes and 392 edges that represents Clermont-Ferrand downtown (cf. Figure 5), with 12 customer demands. Table 1 presents numerical results for the simulation optimization on this instance. In order to test various problem structures, the number of control points in a geodesic varies. We also consider several maximal length T for the bus route. The tabu search has been implemented with C++ and the simulation model with the B++ Simulator framework³. The tests have been performed on a Pentium Centrino 1.7 GHz with G++ 3.2 compiler.

³Information available at http://frog.isima.fr/bruno/?doc=bpp_library+ch=simulator.

Control Points	Length Bound T	Best Evaluation $f_s(x_s^*)$	Number of Evaluations	CPU Time (hours)
5	700000	10.33	3793	5h04
5	800000	10.30	5203	7h04
5	900000	10.32	4807	6h08
5	1000000	10.32	7573	9h44
7	700000	10.32	8201	10h44
7	800000	10.34	6092	8h16
7	900000	10.32	6749	8h56
7	1000000	10.31	7883	10h28
10	700000	10.32	8843	11h20
10	800000	10.33	9706	10h12
10	900000	10.31	10732	13h52
10	1000000	10.34	8812	11h04
15	700000	10.31	6074	8h20
15	800000	10.32	8759	10h08
15	900000	10.33	10699	11h16
15	1000000	10.33	10302	11h08

Table 1: Simulation optimization numerical results.

Execution times can be more than a day. As our purpose here is only to know good practical solutions to compare with model enhancement's ones, we choose to limit the number of diversifications. Notice that the number of simulation evaluations to solve (P_s) is high, and without our restriction, it can reach more than 100000 evaluations. Hence, the solutions provided here are not always the best ones that simulation optimization can provide.

4.3 Straightforward Optimization

Our bus routing problem is strongly related to the well-known Vehicle Routing Problem (VRP) class. Several straightforward optimization techniques can be considered to solve the problem [8]. Especially, [9] propose a mixed integer programming formulation. They show that the tabu search (as defined in the previous section) can provide solutions very close to the optimal solution (when known) or to the best known solution (when it takes too much time or memory to get an optimal solution using mixed integer programming).

Thus, we prefer to choose the tabu search as our method m to solve the optimization problem (P_o). It still searches a good solution x , but this time in the space of solutions C_o . With mixed integer programming, C_o would be larger than C_s ($C_o \supset C_s$), because instead of forcing the solution to be a geodesic, it would look for any directed cycle. With tabu search, it is easier to keep the geodesic structure, so $C_o = C_s$.

Λ_o is an approximation of Λ_s . The implicit constraints of simulation that define Λ_s are replaced by constraints to estimate the time of travel t_d of the customers with demand d . Other constraints fix the waiting times w_i to zero for any bus stop i , due to the assumption that there will be enough buses on the bus route for the waiting times to be insignificant. To sum up, Λ_o approximates the way to estimate the travel times and sets the waiting times to zero.

Under the same conditions than problem (P_s), tests have been performed for (P_o). The problem is solved with tabu search. Table 2 shows the theoretical evaluation $f_o(x_o^*)$ of the best solutions x_o^* found for (P_o), and their simulation evaluation $f_s(x_o^*)$. The resolution is quite fast (around 10 seconds) compared to simulation optimization (several hours), but the solutions do not always have good simulation evaluations. Inside parenthesis is indicated the relative difference $(f_s(x_o^*) - f_s(x_s^*)) / f_s(x_s^*)$.

Control Points	Length Bound T	Straightforward Optimization		Simulation Optimization	CPU Time (seconds)
		Theoretical Eval. $f_o(x_o^*)$	Simulation Eval. $f_s(x_o^*)$	Best Evaluation $f_s(x_s^*)$	
5	700000	11.99	10.29 (-0.4%)	10.33	5.0
5	800000	11.99	10.25 (-0.5%)	10.30	5.0
5	900000	11.99	9.62 (-6.8%)	10.32	5.1
5	1000000	11.99	10.08 (-2.4%)	10.32	5.3
7	700000	11.99	10.31 (-0.1%)	10.32	6.8
7	800000	11.99	9.92 (-4.1%)	10.34	7.1
7	900000	11.98	9.79 (-5.1%)	10.32	7.5
7	1000000	12.00	9.43 (-8.5%)	10.31	7.6
10	700000	11.97	10.18 (-1.4%)	10.32	7.7
10	800000	11.99	9.92 (-4.0%)	10.33	8.9
10	900000	11.99	10.05 (-2.5%)	10.31	9.4
10	1000000	12.00	9.33 (-9.8%)	10.34	10.3
15	700000	11.99	10.29 (-0.2%)	10.31	8.0
15	800000	11.99	10.01 (-3.0%)	10.32	9.0
15	900000	11.99	9.85 (-4.6%)	10.33	9.8
15	1000000	12.00	9.56 (-7.5%)	10.33	10.8

Table 2: Straightforward optimization numerical results.

4.4 Model Enhancement

Simulation optimization provides good practical solutions, but needs a very long time to execute. At the opposite, straightforward optimization is very fast, but provides poor practical solutions. We describe now the model enhancement heuristic (as proposed in Section 4) applied to our bus routing problem, in order to improve the constraints defining Λ_o so problem (P_o) provides good practical solutions.

In problem (P_o) , we assumed that the waiting times w_i are set to zero at any bus stop i . These constraints may make Λ_o a poor approximation of Λ_s . As the method used to solve (P_o) can deal with any constant waiting times $w_i \neq 0$, we propose to modify these theoretical waiting times at each iteration of model enhancement (in order to make Λ_o a better approximation of Λ_s).

Let us denote w_i^k the waiting times for problem (P_o) at iteration k , and $\lambda_s = (w t)$, composed of vectors $w = (w_i)_{i \in V}$ (the estimated waiting times) and $t = (t_d)_{d \in D}$ (the estimated travel times), the simulation evaluation of the current theoretical solution x_o^* . If $w_i \neq 0$, that means i was effectively used as a bus stop during the simulation. Thus, we propose to tend w_i^{k+1} to w_i as follows: $w_i^{k+1} \leftarrow w_i^k + \Delta(w_i - w_i^k)$. If $w_i = 0$, that means i is a vertex where customers never stop. However, we propose to tend w_i^{k+1} to the mean M of the waiting times (estimated since the start of the algorithm) as follows: $w_i^{k+1} \leftarrow w_i^k + \Delta(M - w_i^k)$.

$\Delta < 1$ is a progression step that needs to be tuned. One can choose Δ either constant, or variable according to the number of iterations (in order to ensure some convergence). For this routing problem, we chose a constant $\Delta = 0.1$ and decide to stop the model enhancement after $n = 100$ iterations, or when there is some convergence, i.e. the difference between the theoretical evaluation $f_o(x_o^*) = g(x_o^*, \lambda_o^*)$ and the simulation evaluation $f_s(x_o^*) = g(x_o^*, \lambda_s)$ is less than $\varepsilon = 0.01$. Algorithm 2 summarizes the model enhancement heuristic for our bus routing problem.

Under the same conditions than problems (P_s) and (P_o) , tests have been performed for model enhancement. Table 3 indicates the theoretical evaluation of the best solutions x_e^* found. It shows that it is very close to the simulation evaluation of the same solution. In parenthesis, it indicates the relative improvement of the practical quality of the solution compared to straightforward optimization: $(f_s(x_e^*) - f_o(x_o^*)) / f_s(x_o^*)$. The number of iterations of the model enhancement process is also provided.

Algorithm 2 *Model enhancement heuristic for bus routing problem.*

```

 $k \leftarrow 0$ ; for each vertex  $i \in V$  do  $w_i^k \leftarrow 0$ ;

repeat
   $k \leftarrow k + 1$ ;
  solve  $(P_o)$  with waiting times  $w_i^k, i \in V$ ;
  let  $(x_o^*, \lambda_o^*)$  be an optimal solution of  $(P_o)$ ;
  let  $\lambda_s = (w, t)$  be the simulation evaluation of  $x_o^*$ ;
  update mean  $M$  of the estimated waiting times;

  for each vertex  $i \in V$  do
    if  $w_i \neq 0$  then  $w_i^{k+1} \leftarrow w_i^k + \Delta(w_i - w_i^k)$ ;
    else  $w_i^{k+1} \leftarrow w_i^k + \Delta(M - w_i^k)$ ;
  end for;
until  $|g(x_o^*, \lambda_o^*) - g(x_o^*, \lambda_s)| < \varepsilon$  or  $k \geq n$ ;

 $x_e^* \leftarrow x_o^*$ ;

```

The resolution is slower than straightforward optimization, but is really faster than simulation optimization. In fact, few calls to simulation evaluation are required, contrary to simulation optimization that needs a large amount of evaluations. The quality of the solution provided by model enhancement is usually close to the one provided by simulation optimization, and always improves the solution provided by straightforward optimization.

Control Points	Length Bound T	Model Enhancement			Straight. Opti. $f_s(x_o^*)$	Simulation Opti. $f_s(x_s^*)$	CPU Time (minutes)
		Theoretical $f_o(x_e^*)$	Simulation $f_s(x_e^*)$	Nb Iter.			
5	700000	10.48	10.48 (+1.8%)	51	10.29	10.33	7'46
5	800000	10.38	10.38 (+1.3%)	51	10.25	10.30	7'47
5	900000	10.20	10.20 (+6.0%)	45	9.62	10.32	6'53
5	1000000	10.23	10.23 (+1.5%)	49	10.08	10.32	7'20
7	700000	10.49	10.49 (+1.7%)	46	10.31	10.32	8'24
7	800000	10.40	10.41 (+4.9%)	46	9.92	10.34	8'33
7	900000	10.21	10.22 (+4.4%)	52	9.79	10.32	9'28
7	1000000	10.21	10.21 (+8.3%)	33	9.43	10.31	6'09
10	700000	10.47	10.47 (+2.8%)	50	10.18	10.32	9'08
10	800000	10.41	10.41 (+4.9%)	33	9.92	10.33	7'01
10	900000	10.12	10.12 (+0.7%)	56	10.05	10.31	12'18
10	1000000	10.05	10.05 (+7.7%)	64	9.33	10.34	15'00
15	700000	10.48	10.49 (+1.9%)	51	10.29	10.31	10'02
15	800000	10.42	10.42 (+4.0%)	44	10.01	10.32	9'17
15	900000	10.22	10.22 (+4.0%)	56	9.85	10.33	12'35
15	1000000	10.06	10.06 (+5.2%)	61	9.56	10.33	14'41

Table 3: Model enhancement numerical results.

Figure 6 shows the evolution and the convergence of both theoretical and simulation evaluations of solution x_o^* at each iteration k of the model enhancement process, for the search of a 10-points geodesic with $T = 1000000$.

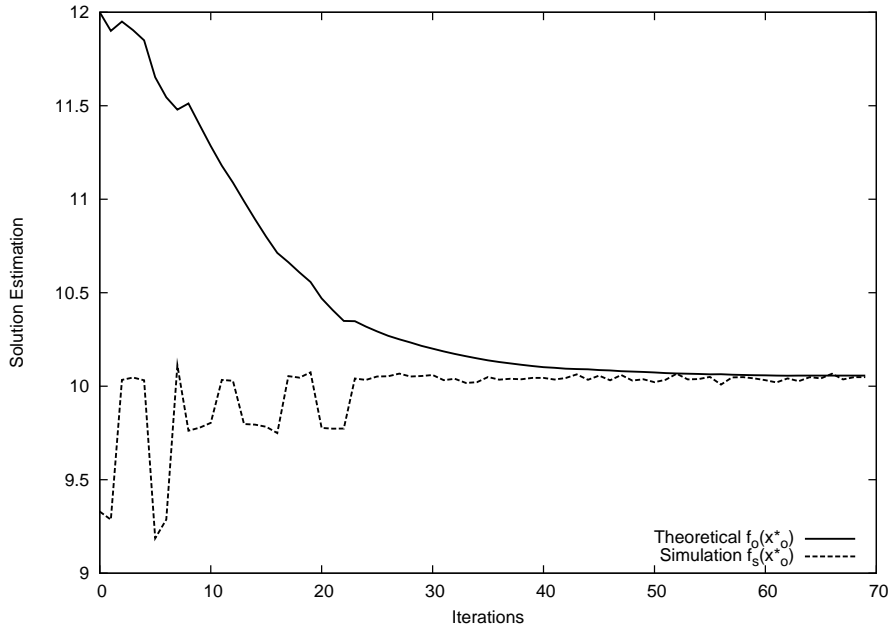


Figure 6: Solution estimation evolution with model enhancement, case 1.

The convergence seems to be achieved quite easily in this case. However, Figure 7 shows the same kind of evolution for the search of a 7-points geodesic with $T = 1000000$. As shown in this figure, the heuristic sometimes oscillates between several good solutions, thus the convergence is more difficult to achieve.

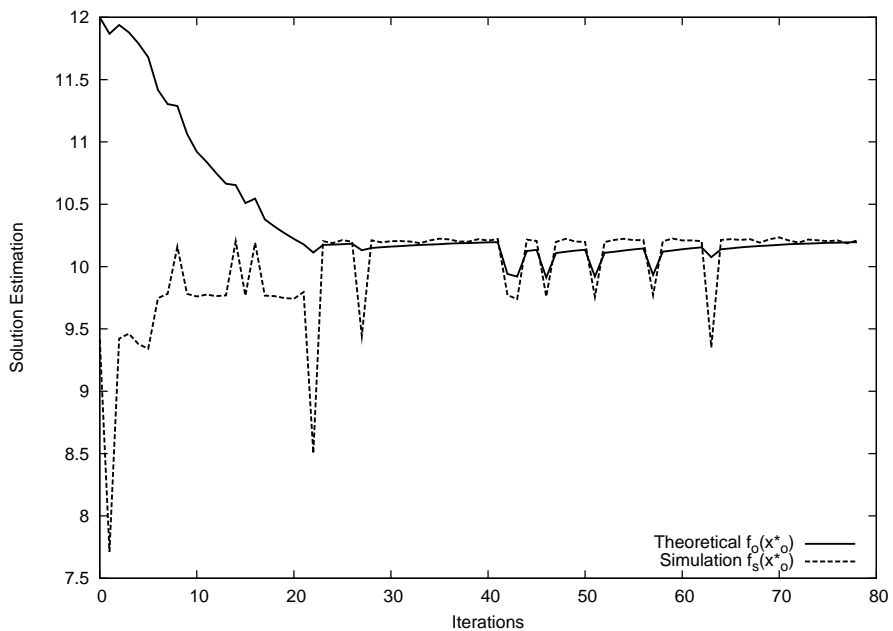


Figure 7: Solution estimation evolution with model enhancement, case 2.

Conclusion

Simulation optimization provides good practical solutions. However, it may take a long time to execute, with lots of simulation evaluations. At the opposite, for some problems, like the routing problem presented here, efficient theoretical approaches can efficiently solve simplified formulations. But, usually, they find solutions that are not so good in practice.

We propose in this article an approach called *model enhancement* that still focuses on the theoretical problem, and tries to improve its formulation in order to take into account practical aspects (estimated by simulation). The final goal is for the theoretical approach to provide good practical solutions.

We formalize model enhancement as problem (P_e) which, in some conditions, should provide the best practical solution (according to simulation evaluation). This problem seems actually very hard to solve. It proposes nevertheless a different way to think of optimization and simulation coupling. Therefore, we present a quite simple heuristic for model enhancement, illustrated on a routing problem. Experimental results show the potential of model enhancement, which needs further investigation.

References

- [1] F. Azadivar. A Tutorial on Simulation Optimization. In *Proceedings of the Winter Simulation Conference*, pages 198–204, 1992.
- [2] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1997.
- [3] A. Herz, E. Taillard, and D. de Werra. Tabu Search. In *Local Search in Combinatorial Optimization*, pages 121–136. Princeton University Press, 1997.
- [4] Dorit S. Hochbaum. *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, 1997.
- [5] Leon S. Lasdon. *Optimization Theory for Large Systems*. MacMillan, 1970.
- [6] Yuri A. Merkurjev and Vladimir K. Visipkov. A Survey of Optimization Methods in Discrete Systems Simulation. In *Proceedings of the European Simulation Symposium on Simulation*, pages 104–110, 1994.
- [7] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, 1999.
- [8] P. Toth and D. Vigo. The Vehicle Routing Problem. In *SIAM Monographs on Discrete Mathematics and Applications*, 2002.
- [9] Loïc Yon, Alain Quilliot, and Christophe Duhamel. Distance Minimization in Public Transportation Networks with Elastic Demands: Exact Model and Approached Methods. In *21st IFIP TC 7 Conference*, 2003.