15th Triennial World Congress, Barcelona, Spain

HYBRID OPTIMAL CONTROL OF MOTORIZED TRAVELING SALESMEN AND BEYOND

Markus Glocker, Oskar von Stryk

Simulation and Systems Optimization Group, Technische Universität Darmstadt, Alexanderstr. 10, 64283 Darmstadt, Germany
Email: {glocker,stryk}@sim.tu-darmstadt.de
http://www.sim.informatik.tu-darmstadt.de

Abstract: Numerical methods for optimal control of hybrid dynamical systems are considered where the discrete dynamics and the nonlinear continuous dynamics are tightly coupled. A decomposition approach for numerically solving general mixed-integer continuous optimal control problems (MIOCPs) is discussed. In the outer optimization loop a branch-and-bound binary tree search is used for the discrete variables. The multiple-phase optimal control problems for the continuous state and control variables in the inner optimization loop are solved by a sparse direct collocation transcription method. A genetic algorithm is applied to improve the performance of the branch-and-bound approach by providing a good initial upper bound on the MIOCP performance index. Results are presented for motorized traveling salesmen problems, new benchmark problems in hybrid optimal control.

Keywords: nonlinear hybrid dynamical systems, mixed-integer optimal control, branch-and-bound, direct collocation transcription, sparse sequential quadratic programming, motorized traveling salesmen, genetic algorithm

1. MIXED-INTEGER OPTIMAL CONTROL PROBLEM

Optimization problems for many technical or chemical processes with hybrid dynamics can be formulated as mixed-integer optimal control problems (MIOCPs).

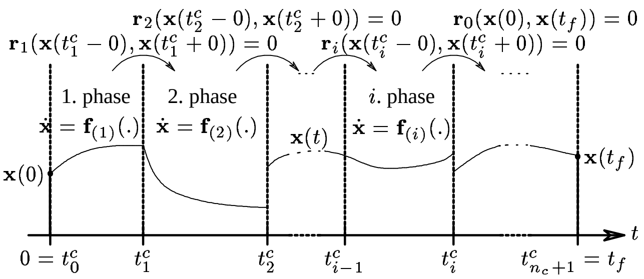

Fig. 1. State dynamics defined in multiple phases, phase transitions through switchings at t_i^c.

A nonlinear dynamical system

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}_{(i)}(\mathbf{x}(t),\mathbf{u}(t),\omega,t),\\ t_i^c \le t &\le t_{i+1}^c, \quad i=0,\dots,n_c\end{aligned} \tag{1}$$

is considered whith the dynamics defined in n_c+1 phases (Figure 1). Here, $\mathbf{x}:[0,t_f]\to\mathbb{R}^{n_x}$ denotes the (piecewise) continuously differentiable state variable, $\mathbf{u}:[0,t_f]\to\mathbb{R}^{n_u}$ denotes the (piecewise) continuous control variable, and $\omega\in\{0,1\}^{n_\omega}$ is a binary control vector. The time t_i^c, $i=0,\dots,n_c$, where the transition from phase i to phase $i+1$ takes place, is usually unknown and must be determined. The transition between two phases (event) is described by switching conditions. The mixed-integer optimal control problem is defined as minimizing the real-valued, hybrid performance index

$$\begin{aligned}J[\mathbf{u},\omega] \;=\; &\sum_{i=1}^{n_c+1}\varphi_{(i)}\left(\mathbf{x}(t_i^c-0),\mathbf{x}(t_i^c+0),\omega,t_i^c\right)\\ &+\sum_{i=1}^{n_c+1}\int_{t_{i-1}^c}^{t_i^c} L_{(i)}(\mathbf{x}(t),\mathbf{u}(t),\omega,t)\,\mathrm{d}t\end{aligned} \tag{2}$$

with respect to the continuous control variable $\mathbf{u}$ and the discrete control vector ω. The solution is subject to further equality and inequality constraints

$$0 \le g_{(i),j}(\mathbf{x}(t), \mathbf{u}(t), \omega, t), \quad j = 1, \dots, n_{g(i)}, \tag{3}$$
$$0 = h_{(i),j}(\mathbf{x}(t), \mathbf{u}(t), \omega, t), \quad j = 1, \dots, n_{h(i)}. \tag{4}$$

Numbers $n_{g(i)}, n_{h(i)}$ and types of constraints in a phase may be different from another phase depending on the actual value of ω. Additionally implicit switching conditions

$$0 = r_{(0),j}(\mathbf{x}(0), \mathbf{x}(t_f), \omega, t_f), \quad j = 1, \dots, n_{r_{(0)}}, \tag{5}$$
$$0 = r_{(i),j}(\mathbf{x}(t_i^c - 0), \mathbf{x}(t_i^c + 0), \omega, t_i^c), \quad j = 1, \dots, n_{r_{(i)}},\ i = 1, \dots, n_c \tag{6}$$

may hold at initial, switching and final times. Furthermore, explicit phase transition conditions (jump maps)

$$\begin{array}{rclrcl} x_j(0) & = & x_{0,j}, & x_k(t_f) & = & x_{f,k}, \\ x_l(t_i^c + 0) & = & R_{(i),l}(\mathbf{x}(t_i^c - 0), \omega, t_i^c). \end{array} \tag{7}$$

may hold. Hereby j, k, l are elements from subsets of $\{1, 2, \dots, n_x\}$ and $x_{0,j}$, $x_{f,k}$ are given real constants.

Finally, linear constraints may be imposed on the binary control vector

$$\mathbf{l}_{\min} \le \mathbf{A}\omega \le \mathbf{l}_{\max}, \quad \mathbf{A} \in \mathbb{R}^{n_A \times n_\omega},\ \mathbf{l}_{\min}, \mathbf{l}_{\max} \in \mathbb{R}^{n_A} \tag{8}$$

For example, if the binary control vector ω steems from a transformation of the discrete states and controls of a general hybrid dynamical system (Buss *et al.*, 2000) the linear constraints describe the feasible transitions of a discrete state or control variable at a switching point. It should be noted that the linear inequalities can be solved independently as a feasibility test for the binary control vector during the outer optimization loop.

2. DECOMPOSITION USING BRANCH AND BOUND

An "continuous" optimal control problem defined in multiple phases remains for each fixed, feasible binary control vector. A naive solution approach to the MIOCP is to enumerate the whole feasible discrete control space and to solve all of the related optimal control problems. However, even for small dimensions this approach is not very efficient and intractable of large dimensions, because MIOCPs may often be NP-complete, as it is the case for the motorized traveling salesman problem (von Stryk and Glocker, 2001).

To potentially avoid the explicit enumeration of the entire discrete control space $\{0, 1\}^{n_\omega}$ a branch-and-bound (B&B) method is used. The idea is to decompose the discrete control space into appropriate subsets. By maintaining lower and upper bounds on the MIOCP performance index, decisions are made wether one of the subsets contains the solution of the MIOCP or not. A subset must not be examined any more, if the solution is not part of it. Otherwise a further subdivision is done and branching is performed in the subset resulting in a binary tree search.

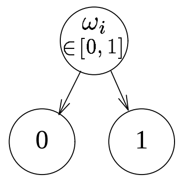

Fig. 2. Branching at an inner node of the binary search tree.

Each node of the binary search tree represents a multiple-phase optimal control problem (OCP). At an inner node of the tree some of the components of ω are fixed to 0 or 1, the other components are free and correspond to a subset of $\{0, 1\}^{n_\omega}$. A lower bound on the performance index of all OCPs corresonding to this subset (and corresponding to all nodes of the subtree having this node as root) can be determined by (globally) solving an OCP corresponding to the node where the free components i of ω are relaxed: $0 \le \omega_i \le 1, \omega_i \in \mathbb{R}$. If the lower bound of an inner node, i.e., the global solution of the corresponding multiple-phase OCP, is larger than the current upper bound of the MIOCP, the solution can not be in this subset. An upper bound of the MIOCP is provided by the performance index of any feasible point at any feasible leaf of the tree. At the leafs all components of ω are set to 1 or 0.

To branch the tree at an inner node, a free variable of the binary control vector ω is chosen and fixed to 0 (or 1, resp.). Various general strategies exist for selecting a branch variable, e.g., first free variable or maximum fractional part. It is not clear which one may lead to highest efficiency of the algorithm for which class of MIOCPs.

Currently, no numerical method exists that solves OCPs of medium-scale dimensions subject to nonlinear dynamics and state constraints guaranteed to the global optimum. Usually, only a local minimum can be found. Thus, the computed performance index of the relaxed MIOCP at an inner node does not necessarily provide a true lower bound. In such a case, a wrong decision is made and the subtree including the solution of the MIOCP may be cut off. As a remedy further relaxations of the continuous problem may be done in principle to achieve convex underestimating optimization problems.

Whenever a node is fathomed it must be decided where to proceed the tree search. Various different strategies exist, e.g., depth first or breadth first search. It is not known a priori which strategy will perform best for which type of MIOCPs. However, for combinatorial optimizers this will not be surprising, e.g. (Cook *et al.*, 1998; Lawler *et al.*, 1985). There are no methods or heuristics that can efficiently applied to most combinatorial optimization problems, as, e.g., (Quasi-) Newton-Methods for continuously differentiable optimization problems. Instead there are almost as many methods and heuristics as problem classes.

However, the overall efficiency of the B&B algorithm strongly depends on good lower and upper bounds. A good initial upper bound is especially crucial for the performance of a B&B algorithm. If the tree search is started at the root without an upper bound, it takes at least n_ω steps to reach a leaf of the tree and get an upper bound. Thus, a search among the leafs should be done first to obtain a good upper bound before starting the branching procedure.

3. UPPER BOUND BY GENETIC ALGORITHM

Evolutionary algorithms try to simulate the principles of evolution. The search for an optimum is hereby steered by various heuristics. There exist different kinds of evolutionary approaches to solve problems. One of them are genetic algorithms (GA). A general description of genetic algorithm can be found, e.g., in (Goldberg, 1989; Michalewicz, 1992)

The aim here is to find a feasible binary control vector, which provides a good (i.e., low) upper bound. Out of a feasible subset of $\{0,1\}^{n_\omega}$, a *selection* of some elements, called parents, is performed. New elements can be achieved by a *recombination* of the parents. Due to a *mutation* of these new elements, the children are generated. The mutation shall bring new material into the population and beware of stagnation. If this is done by proper methods, the children are convenient candidates for a good bound. As an *evaluation* of the children, the optimal index of the related OCP is computed. If one of the children provides a solution that serves as an upper bound which is lower than the current best upper bound, this value is updated. This procedure is repeated until some termination criterium is fulfilled.

It is usually not clear how good an actual solution is and how long it takes to get a better one. A lot of possibilities exist to select (e.g. fitness proportional, tournament), to recombine (e.g. one point crossover, order crossover) and to mutate (e.g. inversion, displacement)(see also section 5). If the strategies are chosen properly, the GA supplies quite good results in a short time.

4. SPARSE DIRECT COLLOCATION FOR MULTIPLE-PHASE OPTIMAL CONTROL PROBLEMS

Each subproblem, where all components of ω are fixed to 0 or 1, is a "continuous" multiple-phase optimal control problem obtained from equations (1) - (8). Using the B&B approach sequences of these continuous problems have to be solved. Hence a method is needed that solves these problems as fast, as robust and as globally as possible.

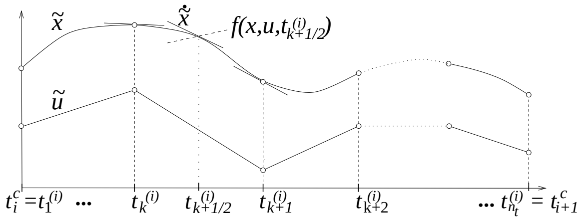

Fig. 3. Direct collocation parameterization of continuous state and control variables.

There are numerical methods based on the Euler-Lagrange differential equations (EL-DEQs) and the Maximum Principle (MP) to solve optimal control problems (Betts, 1998; Buss *et al.*, 2000). These can mainly be divided into two classes: direct and indirect transcription methods (von Stryk and Bulirsch, 1992). The indirect methods approximate a solution by explicitly solving first and second order optimality conditions resulting from the EL-DEQs and the MP. This approach is not flexible enough for the purpose needed here (Buss *et al.*, 2000; von Stryk and Glocker, 2001).

Direct methods are based on a transcription of optimal control problems into (finite dimensional) nonlinearly constrained optimization problems (NLPs). This can be done either by direct shooting or by direct collocation (Betts, 1998; von Stryk and Bulirsch, 1992). Direct methods promise high flexibility and robustness when solving optimal control problems numerically to low or moderate accuracies. Here, the direct collocation approach is potentially faster than direct shooting. This is due to the simultaneous simulation and optimization approach and is only effective if the NLP sparsity is fully utilized. Otherwise the NLP size will severely limit the efficiency.

The transcription of the problem is done by a discretization of the phases of the time interval $[0, t_f]$ in $t^c_i = t^{(i)}_1 < t^{(i)}_2 < \ldots < t^{(i)}_{n^{(i)}_t} = t^c_{i+1}$ (Figure 3) (von Stryk, 1995). On this grid the state variables $\mathbf{x}$ are approximated by piecewise cubic Hermite polynomials $\tilde{\mathbf{x}}(t) = \sum_j \alpha_j \hat{\mathbf{x}}_j(t)$ and the control variables $\mathbf{u}$ by piecewise linear functions $\tilde{\mathbf{u}}(t) = \sum_k \beta_k \hat{\mathbf{u}}_k(t)$. Here, the equations of motion (1) are pointwise fulfilled at the grid points and at their respective midpoints resulting in a set of nonlinear equality constraints (i.e., forming a major part of $\mathbf{a}(\mathbf{y}) = 0$). All inequality constraints of the optimal control problem are to be satisfied at the grid points. All in all a nonlinear, usually non convex, optimization problem

$$\begin{aligned} &\min_{\mathbf{y}} \ \phi(\mathbf{y}) \\ &\text{s.t.} \quad \mathbf{a}(\mathbf{y}) = 0, \ \ \mathbf{b}(\mathbf{y}) \geq 0 \end{aligned} \tag{NLP}$$

is obtained. Here, $\mathbf{y}$ denotes the n_y parameters of the parameterization

$$\mathbf{y} = (\alpha_1, \alpha_2, ..., \beta_1, \beta_2, ..., t_1^c, \dots, t_{n_c}^c, t_f)^T$$

and ϕ the parameterized cost index.

A carefully selected discretization $\tilde{\mathbf{u}}$, $\tilde{\mathbf{x}}$ must satisfy certain convergence properties. One requirement is that the discretized solution must approximate a solution of the EL-DEQs and the Maximum Principle if the grid becomes fine enough, i.e., for $n_t^{(i)} \to \infty$ and $\max\{t_{k+1}^{(i)} - t_k^{(i)} : k = 1, \dots, n_t^{(i)} - 1\} \to 0$, cf. (von Stryk, 1995).

A great advantage of the direct collocation approach is that it provides reliable estimates $\tilde{\lambda}$ of the adjoint variable trajectory along the discretization grid. These estimates are derived from the Lagrange multipliers of the NLP (von Stryk, 1995). They enable a verification of optimality conditions of the discretized solution although the EL-DEQs have not been solved explicitly, e.g., (von Stryk and Glocker, 2001).

Also, local optimality error estimates can be derived that enable efficient strategies for successively refining a first solution on a coarse grid (von Stryk, 1995). Thus, a sequence of related NLPs must be solved whose dimensions increase with the number of grid points.

NLPs can be solved most efficiently numerically by SQP methods. In each SQP iteration a current guess of the solution $\mathbf{y}^*$ is improved by the solution of a quadratic subproblem derived from a quadratic approximation of the Lagrangian of the NLP subject to the linearized constraints (Barclay *et al.*, 1998; Gill *et al.*, 1997). The NLPs resulting from a direct collocation discretization have several special properties:

- The NLPs are of large-scale with very many variables and very many constraints.
- Most of the NLP constraints are active at the solution, e.g., the equality constraints from collocation. Thus, the number of free NLP variables is much smaller than the total number of variables n_y.
- The NLP Jacobians $(\nabla \mathbf{a}(\mathbf{y}), \nabla \mathbf{b}(\mathbf{y}))$ are sparse and structured. Only a few percent of the elements will be nonzero, and the percentage decreases as the number of grid points increases.

These features can fully be utilized by the recently developed large-scale SQP method SNOPT (Gill *et al.*, 1997). The computational speedup achievable by utilizing the NLP structure is more than a factor of one hundred for typical discretized optimal control problems when compared to standard “dense” SQP methods.

Sparse direct collocation methods as for example DIRCOL (von Stryk, 1995; von Stryk and Glocker, 2001) are especially suited for solving the relaxed MIOCPs because of their exceptional robustness and efficiency. Typically only a crude initial guess of parts or all of the solution trajectories of a relaxed MIOCP need to be provided. This robustness is especially useful for the phase transition times which may have quite different positions at the computed solution than as they can be provided initially. The large movements of events from their initial to their final position during the course of the optimization method usually pose high difficulties for other methods. On the other hand, an initially crude solution estimate on a rather coarse discretization grid $n_t^{(i)}$ is not a handicap to such methods but the usual way how the solution procedure begins (von Stryk and Glocker, 2001).

5. THE MOTORIZED TRAVELING SALESMAN

We consider the hybrid dynamical extension of one of the most popular combinatorial optimizations problems: *A motorized salesman is on his way to visit n_c cities exactly once. He does not have to stop in the cities, he just has to drive through them through his journey. He starts at the origin and returns there after his journey. How should he steer and accelerate and in which order should he work off the cities to minimize the overall traveling time?*

In the standard setting as a combinatorial optimization problem, the interconnections between two cities are independent of each other. In the problem setting here, the salesman has to travel on a smooth curve and the performance in between two cities depends on the overall selection of the continuous (steering wheel, gas and brake pedal) and discrete (order of cities) controls.

The motorized traveling salesman (MTSP) can be described by a simplified kinematical model describing a point mass moving in a (x, y)-plane

$$\begin{aligned} \dot{x}(t) &= v_x(t), \quad x(0) = 0 = x(t_f), \\ \dot{y}(t) &= v_y(t), \quad y(0) = 0 = y(t_f), \\ \dot{v}_x(t) &= a_x(t), \ v_x(0) = 0 = v_x(t_f), \\ \dot{v}_y(t) &= a_y(t), \ v_y(0) = 0 = v_y(t_f), \\ a_x^2 &+ a_y^2 \le 7 \ . \end{aligned} \tag{9}$$

Hereby v_x and v_y denote the velocity and a_x, a_y the acceleration or braking of the car in x respectively in y direction, i.e., the continuous state and control variables. The MTSP is formulated as a MIOCP according to Section 1 by $\mathbf{u} = (a_x, a_y)$, $\mathbf{x} = (x, y, v_x, v_y)$ and

$$\min_{\mathbf{u},\, \omega} J[\mathbf{u}, \omega] := t_f + 0.002 \int_0^{t_f} (u_1^2 + u_2^2)\, \mathrm{dt} \tag{10}$$

$$\mathbf{r}_{(i)}(\mathbf{x}(t_i^c - 0), \mathbf{x}(t_i^c + 0), \omega, t_i^c) :=$$

$$\begin{pmatrix} x(t_i^c - 0) \\ y(t_i^c - 0) \end{pmatrix} - \sum_{k=1}^{n_c} \omega_{i,k} \begin{pmatrix} x_k^c \\ y_k^c \end{pmatrix} \tag{11}$$

$$\mathbf{x}(t_i^c + 0) = \mathbf{x}(t_i^c - 0) \tag{12}$$

$$\sum_{k=1}^{n_c} \omega_{i,k} = 1, \quad k = 1, \dots, n_c, \tag{13}$$

$$\sum_{k=1}^{n_c} \omega_{i,k} = 1, \quad i = 1, \ldots, n_c, \tag{14}$$
$$0 \leq \omega_{i,k} \leq 1.$$

At the end of each phase the salesman must visit one of the n_c cities $(x_k^c, y_k^c)^T$. This is ensured by equation (11). The linear constraints make sure, that each city is visited exactly once. Thus the final matrix $\Omega = (\omega_{i,k})_{i,k\in\{1,\ldots,n_c\}} \in \mathbb{R}^{n_c \times n_c}$ has in each column and each row exactly one entry equal to 1. The other values are equal to 0. If $\omega_{i,k} = 1$, the k-th city is visited at the end of the i-th phase. Each tour is a permutation of the

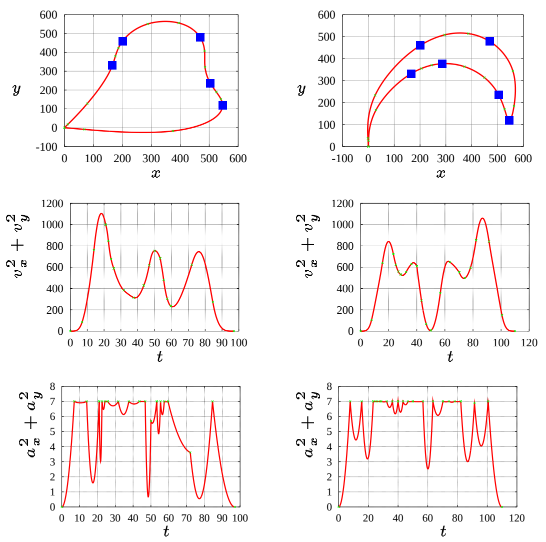

Fig. 4. Solutions of the MTSPs for 5 and 6 cities.

n_c cities. Thus each feasible matrix Ω can be obtained by a permutation of the columns of the identity matrix.

If the salesman has to visit n_c cities, then there are $n_c!$ possible tours, inclusive the symmetric ones. Figure 4 shows two possible tours. In the present formulation n_c^2 binary values are used resulting in a B&B tree with a depth of n_c^2 and a breadth of $2^{n_c^2}$ nodes. All in all the tree has $(2^{n_c^2+1} - 1)$ nodes, but most of them being infeasible with respect to the linear constraints (13), (14).

If a tree search is performed beginning at the root of the tree without the knowledge of an upper bound for the problem, at least n_c^2 nodes have to be analyzed to obtain an initial upper bound. In our numerical experiments, however, even more steps are usually needed to reach the leafs. Thus, the search for an optimum should rather begin at the leafs of the tree until a good feasible solution is provided. The B&B algorithm starts afterwards to proof wether this bound is optimal (in convex cases) or to find a better one.

To find an initial upper bound a GA is used. Hereby a starting population is created. Two tours (parents) out of this population are selected, each by a tournament. At a tournament k individuals of the population are selected at random and the one with the best performance is picked. If k is a small value, individuals with bad performance have a better chance to be elected.

The recombination should take care about generating feasible tours for the TSP. The only thing that should be done is a permutation of the rows (or columns) of Ω. At the moment an order crossover (OX) is used. To demonstrate the procedure, the order crossover is performed for two tours to 10 cities:

Parent A	2	6	**10**	**3**	**8**	5	1	7	4	9
Parent B	**7**	3	**6**	**9**	**1**	8	10	**5**	**2**	**4**

Parent A represents a circular tour from the origin visiting first city 2, then city 6, then city 10 and so on. Randomly a crossover section (e.g., the part of parent A with entries 10 3 8) is taken. The first child $\tilde{A}$ recieves the values of the crossover section of parent A.

Child $\tilde{A}$	x	x	**10**	**3**	**8**	x	x	x	x	x

The free places x are filled with the values of parent B. These values are taken in the order as they are, but without the values alredy contained in $\tilde{A}$.

Child $\tilde{A}$	**7**	**6**	**10**	**3**	**8**	**9**	**1**	**5**	**2**	**4**

$\tilde{A}$ contains part (10 3 8) of tour A and parts (9 1) and (5 2 4) of tour B as they have been. Thus, the subtours are not mixed too much and hopefully the good parts will be bequeathed.

In the last step the two children are permutated. This is done by a simple inversion. Hereby some sequenced rows of the matrix Ω are chosen and replaced in reverse order. So new circular tours are achieved and added to the population.

Table 1 shows some numerical results. The upper half shows the parameters of the GA. If the GA has reached the "maximum population", it stops and the B&B starts. In the last row there is only the entry of one number at "best upper bound". In this case, the upper bound was given by the user for test purposes. The lower half of Table 1 is about the B&B. It shows the next bound, which was found by the algorithm and the number of the node, where the bound was found. The nodes are numberd in the order as they are tested. Finally the total amount of tested nodes by the algorithm to find the solution is given.

Obviously, the number of nodes can be reduced significantly, if a good initial upper bound is given. The results of Table 1 show that a genetic algorithm is a valid approach to obtain such an initial upper bound.

It depends on the location of the cities, how much the performance indeces corresonding to different tours differ. An interesting result can be seen in the last column of Table 1. The bound provided by the user is almost equal to the expected solution, but the B&B search still tests 1058 nodes to find the solution. Thus, tighter lower bounds may be needed to achieve better performance.

Table 1. Numerical results.

number of cities	5	5	6	6	6	7	7	7
GA								
initial population	-	10	-	20	20	30	50	-
max. population	-	20	-	40	60	100	200	-
selection range	-	8	-	15	15	25	40	-
best upper bound	none	109.05	none	116.57	112.88	127.96	117.92	112.86
B&B								
next upper bound	109.05	97.92	140.65	113.79	111.07	124.15	115.0	112.85
computed at node	59	101	85	223	279	213	949	725
overall tested nodes	195	129	11331	825	647	1931	1397	1058
solution	97.92	97.92	111.07	111.07	111.07	112.85	112.85	112.85

The computed solutions of the motorized traveling salesman problems for 5, 6 (the 5 cities plus a new one) and 7 cities (the 6 plus a new one) are displayed in Figures 4 and 5. The population of all actual investigated tours may obtain one tour several times. By varying the starting point for the evaluation of the tours, we observed that a better local solution could be found. The computational times for a single column in Table 1 varies between a few minutes (for 5 cities) and several hours (for 7 cities) on a Pent. III, 900 MHz PC.

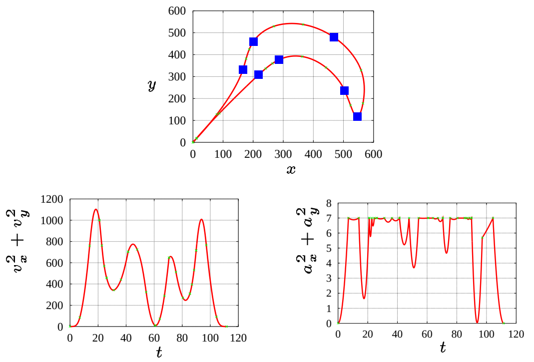

Fig. 5. Solution trajectories for 7 cities.

6. CONCLUSION

A decomposition approach for solving fairly general hybrid optimal control problems (MIOCPs) based on B&B and sparse direct collocation has been developed. It has been successfully applied to several benchmark problems in hybrid optimal control for motorized traveling salesmen. A genetic algorithm has been investigated to obtain a good initial upper bound on the performance index for a more efficient, subsequent B&B tree search leading to a significant reduction in the number of investigated nodes. Ongoing work investigates algorithmic improvements through different strategies for the GA and the B&B method as well as the application of the decomposition approach to further problems such as cooperative teams of mobile robots in robot soccer.

7. REFERENCES

Barclay, A., P.E. Gill and J.B. Rosen (1998). SQP methods and their application to numerical optimal control. In: *Variational Calculus, Optimal Control and Application* (W.H. Schmidt, K. Heier, L. Bittner and R. Bulirsch, Eds.). Vol. International Series of Numerical Mathematic 124. pp. 207–222. Birkhäuser. Basel.

Betts, J.T. (1998). Survey of numerical methods for trajectory optimization. *AIAA J. Guidance, Control, and Dynamics* **21(2)**, 193–207.

Buss, M., O. von Stryk, R. Bulirsch and G. Schmidt (2000). Towards hybrid optimal control. *at-Automatisierungstechnik* **48**, 448–459.

Cook, W.J., W.H. Cunningham, W.R. Pulleyblank and A. Shrijver (1998). *Combinatorial Optimization*. J. Wiley & Sons. New York.

Gill, P.E., W. Murray and M.A. Saunders (1997). SNOPT: An SQP algorithm for large-scale constrained optimization. Report NA 97-2, Math Dept., University of California, San Diego.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

Lawler, E.L., J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys (1985). *The Traveling Salesman Problem*. J. Wiley & Sons. Chichester.

Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer. New York.

von Stryk, O. (1995). *Numerische Lösung optimaler Steuerungsprobleme: Diskretisierung, Parameteroptimierung und Berechnung der Adjungierten Variablen*. VDI-Verlag, Düsseldorf. Fortschritt-Berichte VDI, Reihe 8, Nr. 441.

von Stryk, O. and M. Glocker (2001). Numerical mixed-integer optimal control and motorized traveling salesmen problems. *APII - JESA (Journal europen des systmes automatiss - European Journal of Control)* **35**(4), 519–533.

von Stryk, O. and R. Bulirsch (1992). Direct and indirect methods for trajectory optimization. *Annals of Operations Research* **37**, 357–373.