

Why under-constrained systems are not that bad

Simon E.B. Thierry, Pascal Schreck and Pascal Mathis

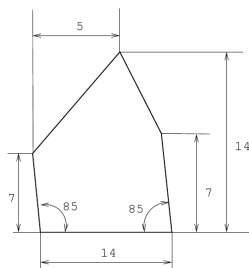
Université de Strasbourg
LSiIT – UMR UdS - CNRS 7005

ADG '10: 8th Workshop on Automated Deduction in Geometry



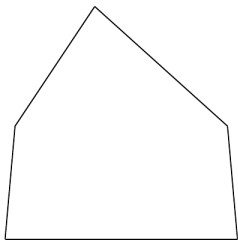
Geometric Constraints Solving

- Input : a set of geometric constraints concerning geometric entities (sketch)
- solutions : figures satisfying the constraints (coordinates)



Geometric Constraints Solving

- Input : a set of geometric constraints concerning geometric entities (sketch)
- solutions : figures satisfying the constraints (coordinates)



Geometric Constraints Solving

- Input : a set of geometric constraints concerning geometric entities (sketch)
- solutions : figures satisfying the constraints (coordinates)
- Application fields :
 - Computer-Aided Design
 - Computer-Aided Education
 - Molecular Chemistry
 - Robotics
 - ...

Solving methods

- algebraic methods : work on equations ;
 - symbolic : manipulate equations
 - numerical : iterative process
- geometric methods : reason on geometry
 - expert systems : explicit construction rules
 - graph-based methods : combinatorial methods
 - lack of generality
- general trend : decomposition

C. M. Hoffmann et R. Joan-Arinyo, *A brief on constraint solving*, 2005

C. Jermann, G. Trombettoni, B. Neveu et P. Mathis, *Decomposition of Geometric Constraint Systems : a Survey*, 2006

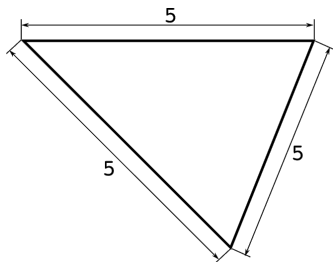
Solving methods

- algebraic methods : work on equations ;
 - symbolic : manipulate equations
 - numerical : iterative process
- geometric methods : reason on geometry
 - expert systems : explicit construction rules
 - graph-based methods : combinatorial methods
 - lack of generality
- general trend : decomposition

C. M. Hoffmann et R. Joan-Arinyo, *A brief on constraint solving*, 2005
C. Jermann, G. Trombettoni, B. Neveu et P. Mathis, *Decomposition of Geometric Constraint Systems : a Survey*, 2006

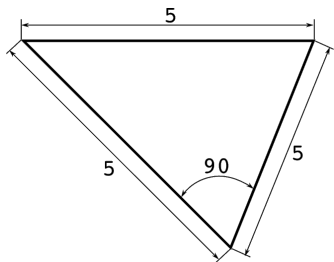
Constrainedness levels

- well-constrained : finite non-zero number of solutions (rigid)
- over-constrained : no solutions
- under-constrained : infinite number of solutions



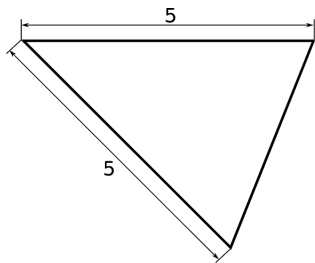
Constrainedness levels

- well-constrained : finite non-zero number of solutions (rigid)
- over-constrained : no solutions
- under-constrained : infinite number of solutions



Constrainedness levels

- well-constrained : finite non-zero number of solutions (rigid)
- over-constrained : no solutions
- under-constrained : infinite number of solutions



Under-constrained systems are an error

Under-constrainedness is an error case : it must be corrected by adding constraints to reach rigidity

Joan-Arinyo et al., Transforming an under-constrained geometric constraint problem into a well-constrained one, 2003 : two issues :

- well-constrained completion (add constraints to an under-constrained GCS so that it becomes well-constrained)
- completion (add constraints in such a way that the new GCS can be solved by geometric constructions)

Zhang and Gao later (2006) added optimal well-constrained completion (add constraints so that the new GCS is well-constrained and the set of equations to solve simultaneously is of minimal size).

The literature contains a great deal of algorithms to complete an under-constrained system and make it rigid.

Under-constrained systems are an error

Under-constrainedness is an error case : it must be corrected by adding constraints to reach rigidity

Joan-Arinyo *et al.*, *Transforming an under-constrained geometric constraint problem into a well-constrained one*, 2003 : two issues :

- well-constrained completion (add constraints to an under-constrained GCS so that it becomes well-constrained)
- completion (add constraints in such a way that the new GCS can be solved by geometric constructions)

Zhang and Gao later (2006) added optimal well-constrained completion (add constraints so that the new GCS is well-constrained and the set of equations to solve simultaneously is of minimal size).

The literature contains a great deal of algorithms to complete an under-constrained system and make it rigid.

Under-constrained systems are an error

Under-constrainedness is an error case : it must be corrected by adding constraints to reach rigidity

Joan-Arinyo *et al.*, *Transforming an under-constrained geometric constraint problem into a well-constrained one*, 2003 : two issues :

- well-constrained completion (add constraints to an under-constrained GCS so that it becomes well-constrained)
- completion (add constraints in such a way that the new GCS can be solved by geometric constructions)

Zhang and Gao later (2006) added optimal well-constrained completion (add constraints so that the new GCS is well-constrained and the set of equations to solve simultaneously is of minimal size).

The literature contains a great deal of algorithms to complete an under-constrained system and make it rigid.

Under-constrained systems are not that bad !

We should solve under-constrained systems the same way we solve rigid systems

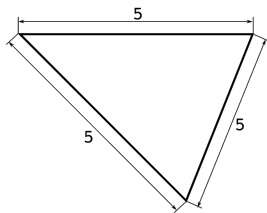
- user intent (pair of scissors, desk lamp, car, ...)
- path towards rigid systems
- needed for incremental methods
- application to dynamic geometry
- ...

Plan

- 1 Introduction : Geometric Constraints Solving
- 2 Under-constrainedness definition
- 3 Under-constrained systems in decomposition
- 4 W-decomposition
- 5 Quasi-decomposition
- 6 Incremental modeling
- 7 Conclusion

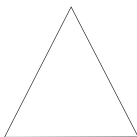
1 lied : rigid and yet under-constrained systems

- Rigid GCS



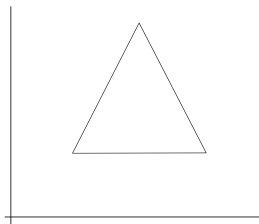
I lied : rigid and yet under-constrained systems

- Rigid GCS



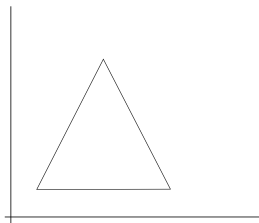
I lied : rigid and yet under-constrained systems

- Rigid GCS



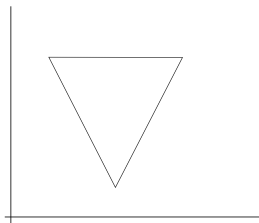
I lied : rigid and yet under-constrained systems

- Rigid GCS with an infinite number of solutions



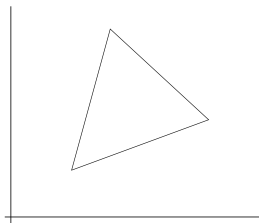
I lied : rigid and yet under-constrained systems

- Rigid GCS with an infinite number of solutions



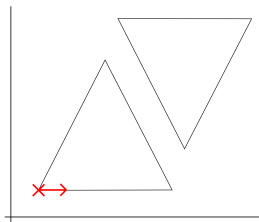
I lied : rigid and yet under-constrained systems

- Rigid GCS with an infinite number of solutions



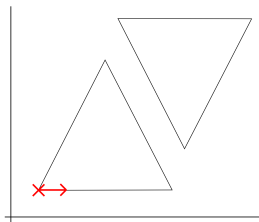
I lied : rigid and yet under-constrained systems

- Rigid GCS with an infinite number of solutions
- two particular solutions – reference
- apply rigid motions to generate the whole solution space



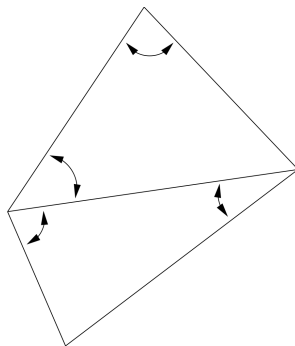
I lied : rigid and yet under-constrained systems

- Rigid GCS with an infinite number of solutions
- two particular solutions – reference
- apply rigid motions to generate the whole solution space
- Well-constrained *modulo* rigid motions

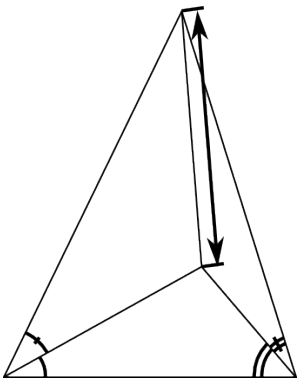


Multi-group approach

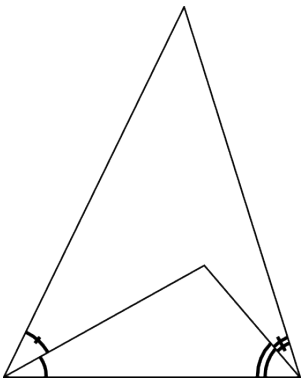
- transformations groups :
 - translations
 - rotations
 - scalings
 - and their compositions
- act globally
- reference for a transformation group



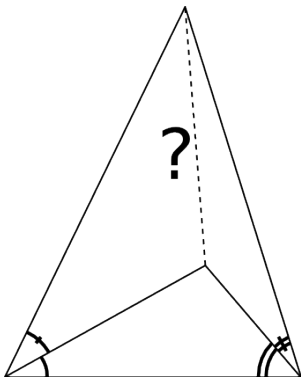
Useful for decomposition



Useful for decomposition



Useful for decomposition



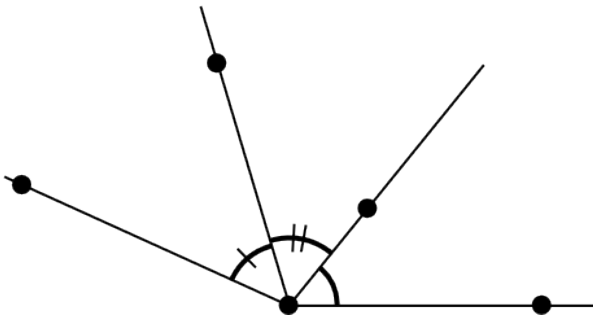
Schreck and Mathis, *Geometrical constraint system decomposition : a multi-group approach*, 2006

Useful for decomposition [2]

van der Meiden and Bronsvort, *A non-rigid cluster rewriting approach to solve systems of 3D geometric constraints*, 2010

Three kinds of clusters :

- 1 classical rigid clusters
- 2 scalable clusters (well-constrained *mod* similarities)
- 3 radial clusters



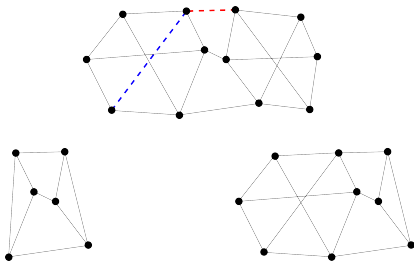
W-decomposition

- Witness interrogation method : Michelucci and Foufou, ADG '06
- allows to identify the maximal rigid subsystems (MRS)
- search for the MRSs of under-constrained subsystems of a rigid system
- remove a constraint, identify MRSs, recurse

Michelucci, Schreck, *et al.*, *Using the witness method to detect rigid subsystems of geometric constraints in CAD*, 2010

W-decomposition

- Witness interrogation method : Michelucci and Foufou, ADG '06
- allows to identify the maximal rigid subsystems (MRS)
- search for the MRSs of under-constrained subsystems of a rigid system
- remove a constraint, identify MRSs, recurse

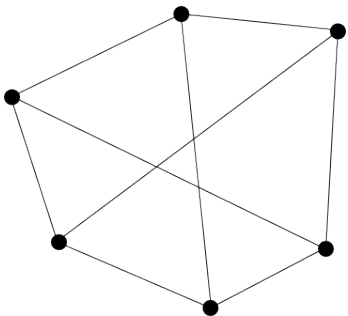


Michelucci, Schreck, *et al.*, *Using the witness method to detect rigid subsystems of geometric constraints in CAD*, 2010

Relax...

Fabre and Schreck, *Combining symbolic and numerical solvers to simplify indecomposable systems solving*, 2008

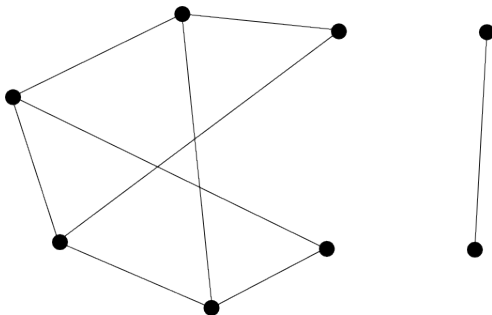
- system you cannot decompose
- but you can solve one of its under-constrained subsystems
- add parameterized constraints
- numerical iterations



Relax...

Fabre and Schreck, *Combining symbolic and numerical solvers to simplify indecomposable systems solving*, 2008

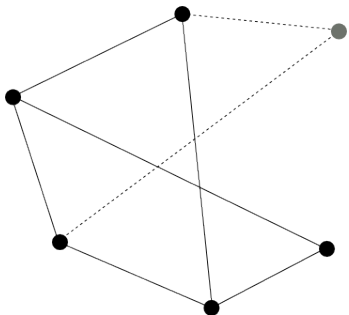
- system you cannot decompose
- but you can solve one of its under-constrained subsystems
- add parameterized constraints
- numerical iterations



Relax...

Fabre and Schreck, *Combining symbolic and numerical solvers to simplify indecomposable systems solving*, 2008

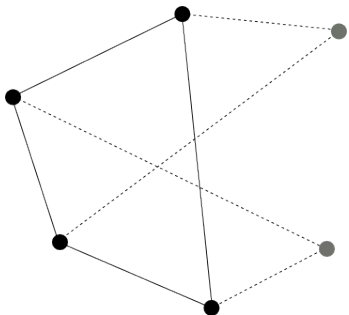
- system you cannot decompose
- but you can solve one of its under-constrained subsystems
- add parameterized constraints
- numerical iterations



Relax...

Fabre and Schreck, *Combining symbolic and numerical solvers to simplify indecomposable systems solving*, 2008

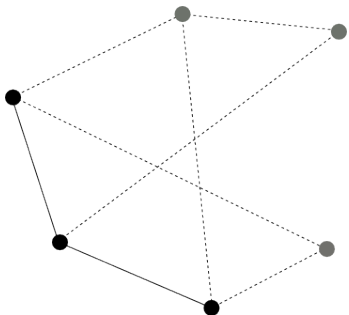
- system you cannot decompose
- but you can solve one of its under-constrained subsystems
- add parameterized constraints
- numerical iterations



Relax...

Fabre and Schreck, *Combining symbolic and numerical solvers to simplify indecomposable systems solving*, 2008

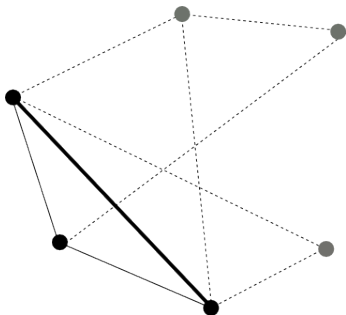
- system you cannot decompose
- but you can solve one of its under-constrained subsystems
- add parameterized constraints
- numerical iterations



Relax...

Fabre and Schreck, *Combining symbolic and numerical solvers to simplify indecomposable systems solving*, 2008

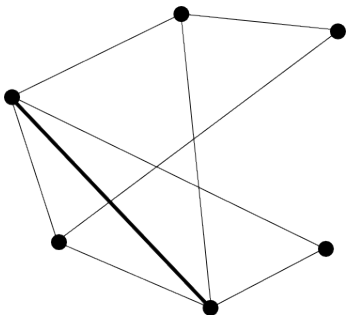
- system you cannot decompose
- but you can solve one of its under-constrained subsystems
- add parameterized constraints
- numerical iterations



Relax...

Fabre and Schreck, *Combining symbolic and numerical solvers to simplify indecomposable systems solving*, 2008

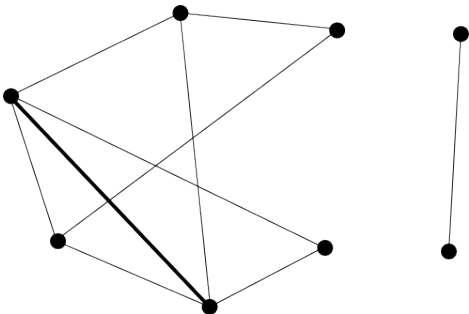
- system you cannot decompose
- but you can solve one of its under-constrained subsystems
- add parameterized constraints
- numerical iterations



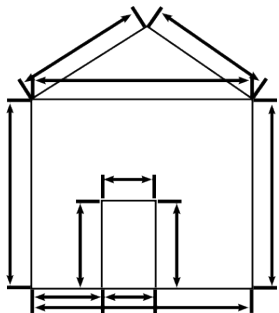
Relax...

Fabre and Schreck, *Combining symbolic and numerical solvers to simplify indecomposable systems solving*, 2008

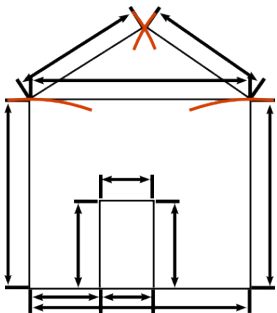
- system you cannot decompose
- but you can solve one of its under-constrained subsystems
- add parameterized constraints
- numerical iterations



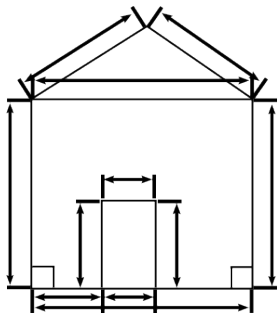
Non-expert users and under-constrainedness



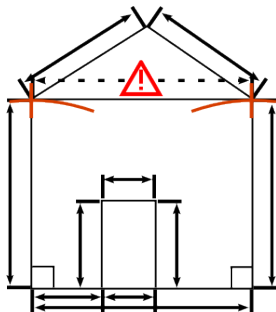
Non-expert users and under-constrainedness



Non-expert users and under-constrainedness

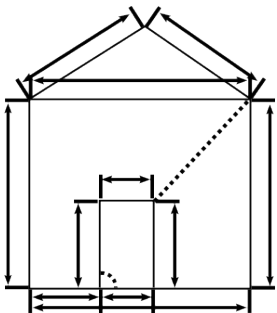


Non-expert users and under-constrainedness



Over-constrained : distance cannot be generically satisfied

Non-expert users and under-constrainedness

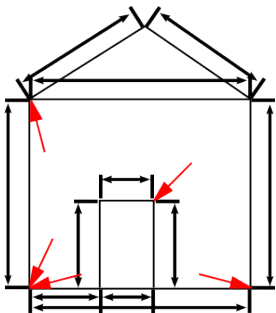


Parameterization

The other approach : tell the user which geometric elements should be anchored.

- compute a parameterization
- compute appropriate values for the parameters
- compute solution figures

Enables trial and error modeling.

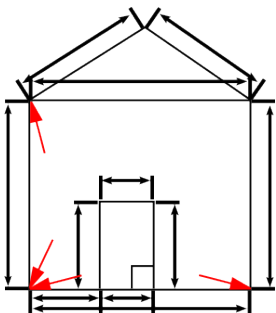


Parameterization

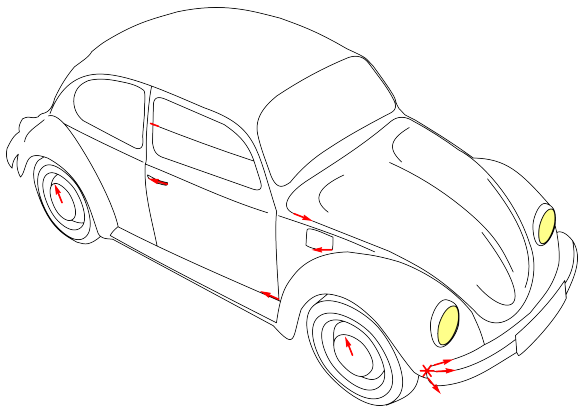
The other approach : tell the user which geometric elements should be anchored.

- compute a parameterization
- compute appropriate values for the parameters
- compute solution figures

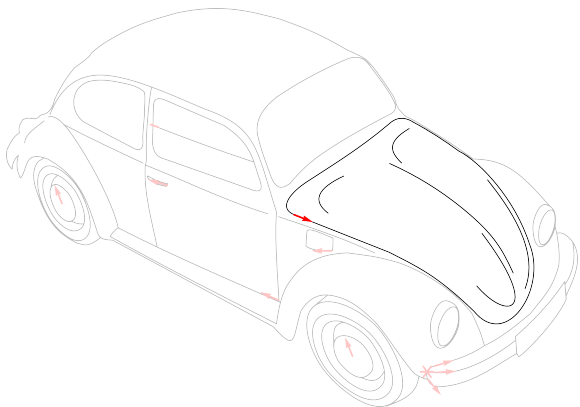
Enables trial and error modeling.



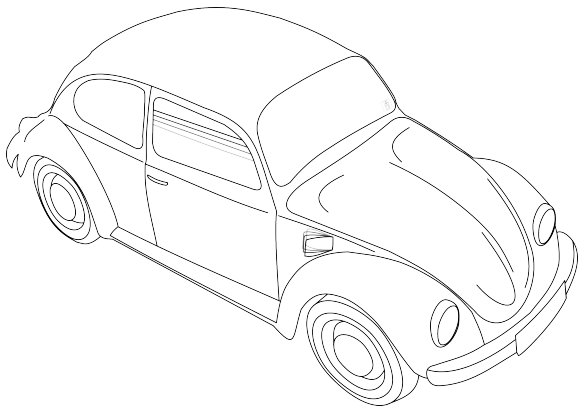
Accessible user interfaces



Accessible user interfaces



Accessible user interfaces



Conclusion

Under-constrained geometric constraint systems should not be considered as design errors

- using under-constrained systems helps solving rigid systems
- rigid systems and under-constrained systems should be completed the same way : by anchoring them in the plane/space
- leads to accessible constraint-based design softwares
- several ways to browse the solution space
 - change the values of a given parameter
 - automatically animate the solutions
 - select the subset of the solutions which satisfy an additional condition : add a constraint

Thanks for your attention